NASA Technical Memorandum

NASA TM -86593

AUTOMATIC DETECTION OF ELECTRIC POWER TROUBLES

(AI Application)

By Caroline Wang, Hugh Zeanah, Audie Anderson, and Clint Patrick

Software and Data Management Division
Information and Electronic Systems Laboratory
Science and Engineering Directorate

April 1987

(NASA-TM-86593) AUTOMATIC DETECTION OF ELECTRIC POWER TROUBLES (AI APPLICATION) (NASA) 60 p Avail: NTIS HC A05/MF A01 CSCL 09C

N87-23893
Unclas
H1/33 0078999

National Aeronautics and
Space Administration

George C. Marshall Space Flight Center

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO.	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
NASA TM - 86593		

4. TITLE AND SUBTITLE	5. REPORT DATE
Automatic Detection of Electric Power Troubles (AI Application)	April 1987
	6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S)	8. PERFORMING ORGANIZATION REPORT #
Caroline Wang, Hugh Zeanah, Audie Anderson, and Clint Patrick	
9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. WORK UNIT NO.
George C. Marshall Space Flight Center Marshall Space Flight Center, Alabama 35812	11. CONTRACT OR GRANT NO.
	13. TYPE OF REPORT & PERIOD COVERED
12. SPONSORING AGENCY NAME AND ADDRESS	Technical Memorandum
National Aeronautics and Space Administration Washington, D.C. 20546	14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

Prepared by Software and Data Management Division, Information and Electronics Systems Laboratory, Science and Engineering Directorate.

16. ABSTRACT

The design goals for the Automatic Detection of Electric Power Troubles (ADEPT) were to enhance Fault Diagnosis Techniques in a very efficient way. ADEPT system was designed in two modes of operation: (1) Real Time Fault Isolation and (2) a local simulator which simulates the models theoretically.

17. KEY WORDS	18. DISTRIBUTION STATEMENT
Automatic Detection of Electric Power Troubles Fault Isolation Expert System	Unclassified – Unlimited

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
Unclassified	Unclassified	83	NTIS

MSFC - Form 3292 (May 1969)

For sale by National Technical Information Service, Springfield, Virginia 22161

ACKNOWLEDGMENTS

I wish to thank a number of people who helped me get the Automatic Detection of Electric Power Troubles system operationally successful. In particular:

Mr. Ed New, from Kennedy Space Center who helped me understand the Knowledge-based Automatic Test Equipment (KATE).

Mr. Hugh Zeanah, my team leader, who did the analysis and design of the ADEPT rule system.

Mr. Audie Anderson, our Technical Advisor, who advised me on future plans for this project.

Mr. Clint Patrick, our in-house hardware engineer, who modified and enhanced the breadboard, and who also helped in the documentation.

Mr. Steve Espy, from Martin Marietta, who helped us in our understanding of the breadboard.

Mr. Steve Davis, the student from The University of Alabama in Huntsville, who also provided some help.

I especially appreciate all the support from Mr. Dave Weeks of the Electrical Division, Information and Electronic Systems Laboratory of Marshall Space Flight Center, and Mr. John Jamieson of Kennedy Space Center.

I also wish to thank our managers for their support.

TABLE OF CONTENTS

TECHNICAL MEMORANDUM

AUTOMATIC DETECTION OF ELECTRIC POWER TROUBLES (AI Application)

I. INTRODUCTION

Marshall Space Flight Center (MSFC) is involved with the design and development of the Automation of Electrical Power System. The design goals were to enhance Fault Diagnosis Techniques and to develop Fault Recovery Techniques.

The design of the Automatic Detection of Electrical Power Troubles (ADEPT) system includes:

(1) The Real Time Fault Isolation through a breadboard which models the power components.

(2) A local simulator which simulates the models theoretically.

II. ADEPT HISTORY BACKGROUND

In 1985, Martin Marietta delivered to MSFC the Fault Isolation Expert System (FIES II) which was implemented in Automated Reasoning Tool (ART) on the Symbolics 3670. A two rack, 350 watt, 3 channel, electrical power system breadboard was also included.

MSFC was experimenting with other software techniques to improve the performance and speed. ADEPT was built with the MSFC rule system and utilized the existing FIES II breadboard and FIES II software interface. ADEPT also uses the Knowledge-based Automatic Test Equipment (KATE) which was developed by Kennedy Space Center as a tool for the simulation version.

The real time Fault Isolation version was implemented in LISP which can quickly search the fault, display the fault data, and print out the reasons. The simulation version can help in theoretical study.

In 1987, The University of Alabama in Huntsville was also involved with MSFC via a contract for study and help on this project.

III. PURPOSE

The purpose of this project is:

1. To enhance fault diagnosis and recovery techniques in a very efficient way.

2. To enable multi-use of the existing software for other problems and to develop existing software for multi-use.

3. To determine the advantages and disadvantages of using high level commercial software tools.

The purpose of this report is:

1. To describe, for the engineers' uses as a reference manual, the:

   (a) Automatic Detection of Electrical Power System

   (b) Basic software and hardware system layout

   (c) Generic rule system

   (d) Knowledge base

   (e) The user's guide

   (f) Function descriptions

2. To demonstrate the current accomplishments and future plans for our manager's information.

IV. ADEPT SYSTEM OVERVIEW

A. Flow Diagram

Figures 1, 2, and 3 are flow diagrams of the ADEPT system.

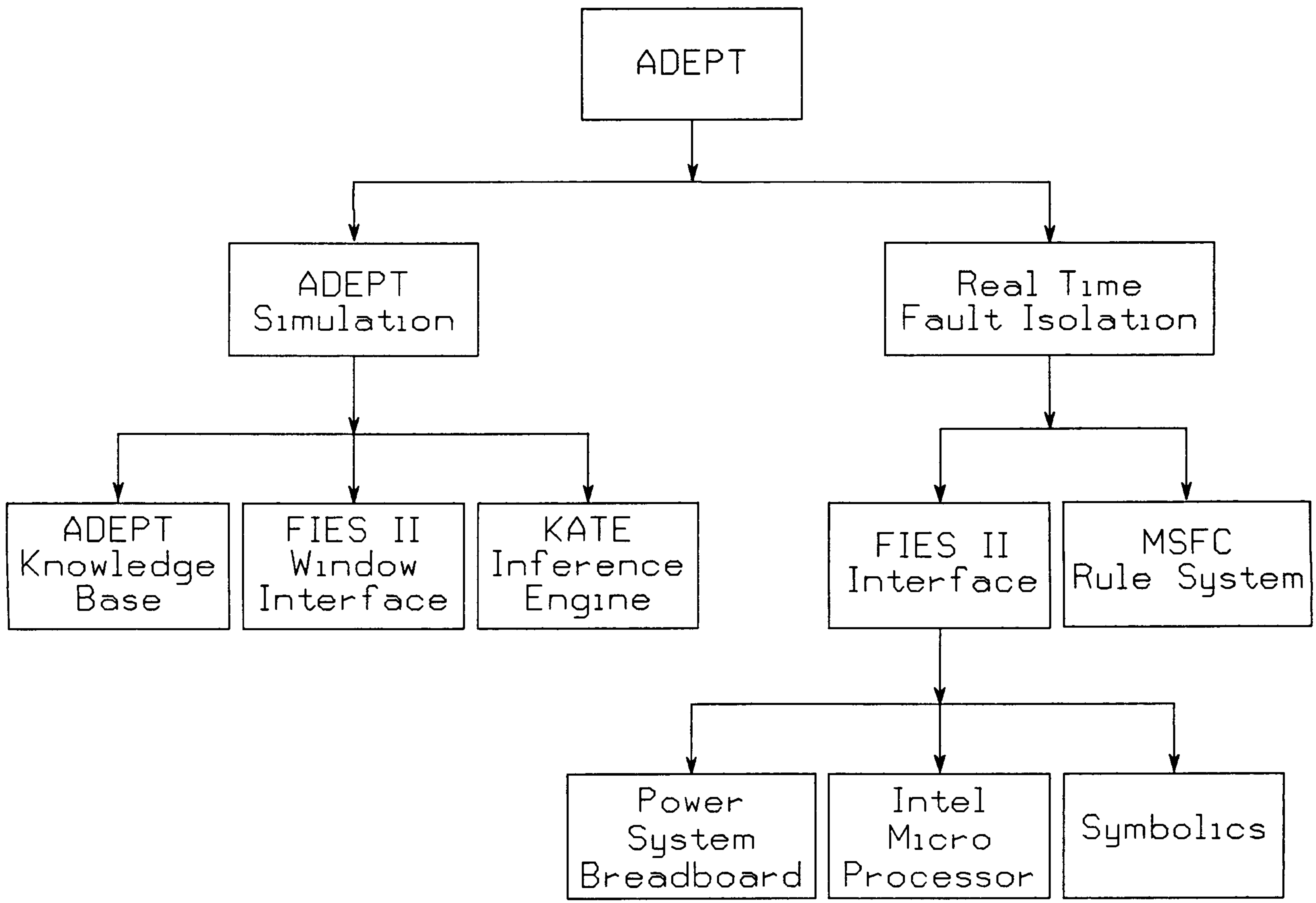


Figure 1. ADEPT system flow diagram.

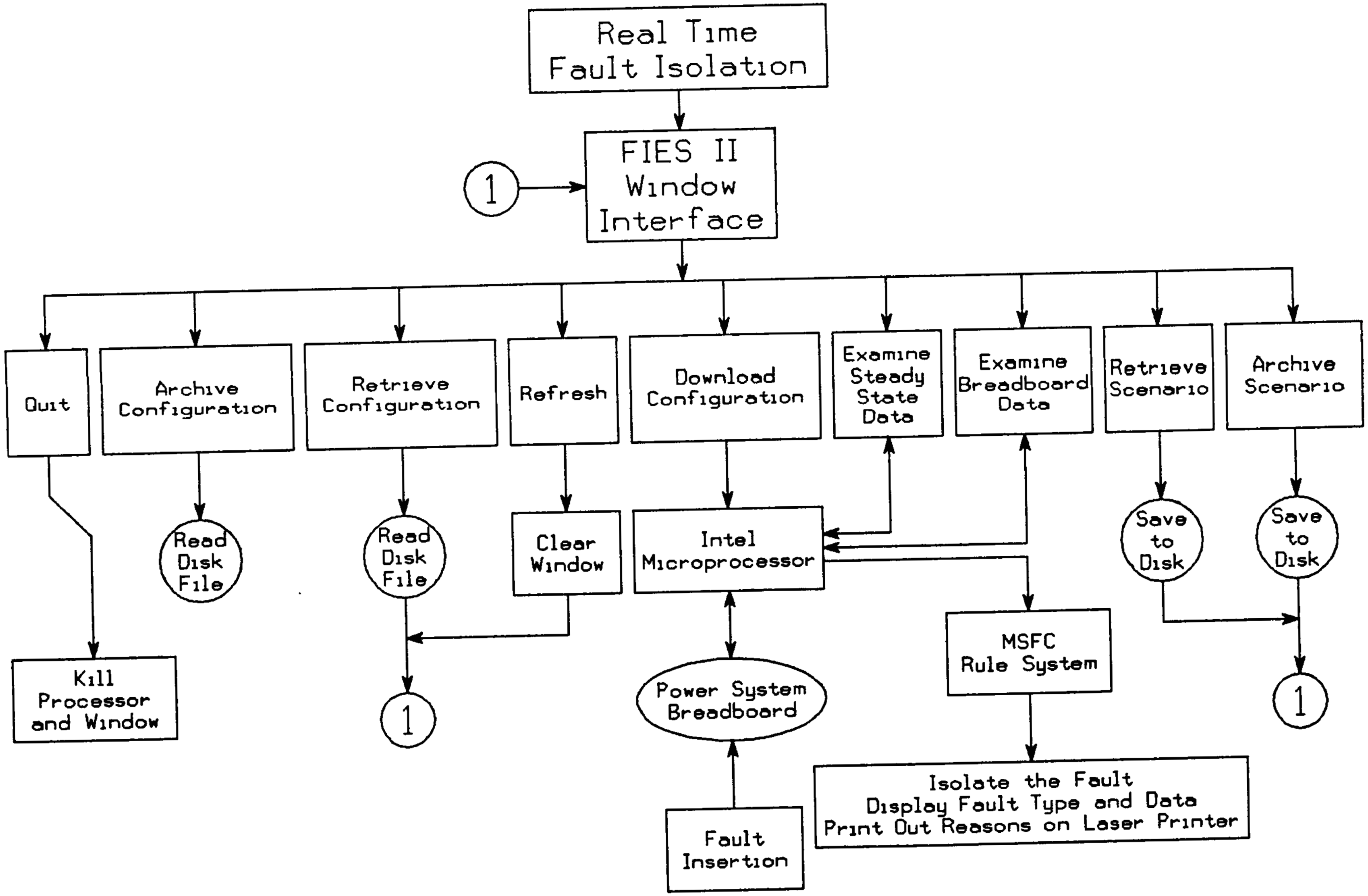


Figure 2. ADEPT real-time fault isolation flow diagram.

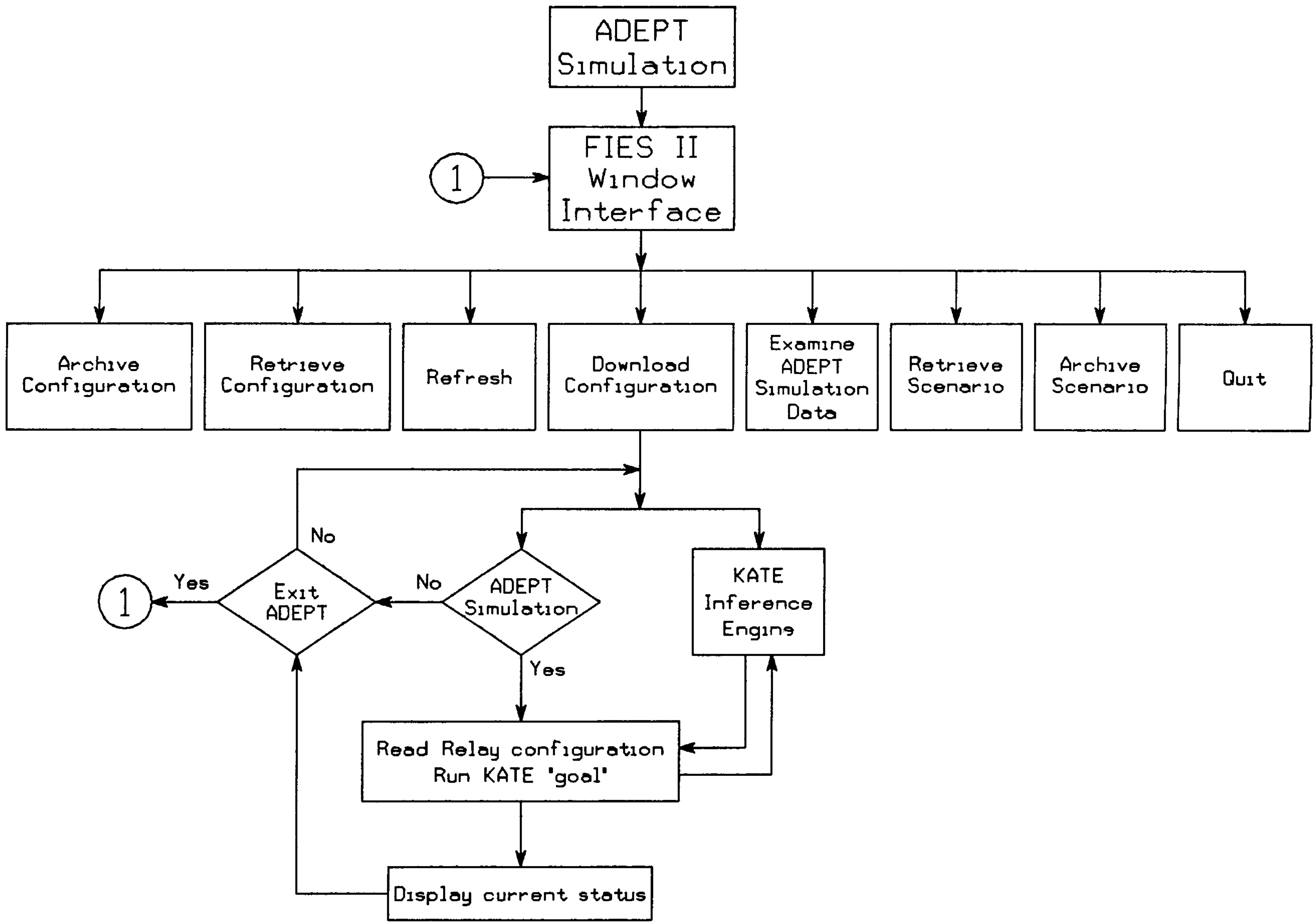


Figure 3. ADEPT simulation flow diagram.

B. Basic Software Layout

1. ADEPT System and Package

The following pages show the ADEPT system and package basic software layout.

ADEPT integrates software from three different suppliers to offer an advanced fault detection system, and is designed for two modes of operation:

1.) Real time fault isolation of components on the power system breadboard through FIES II inteface (Developed by Martin Marietta, Denvor) is made possible with an in-house-developed rule system. Faults are quickly detected, display, and reasoning is provided.
FIES II interface was written in Zetalisp with User Package.
MSFC rules system was written in Common Lisp with User Package.

2.) A simulated version, used for theoretical studies, is implemented using a modified version of KATE (Kenney Space Center's Knowledge Based Automatic Test Equipment), FIES II window interface, and ADEPT knowledge base.

KATE Inference Engine was written in Common Lisp with KATE and KATE-Simulate Packages.

ADEPT combined all the systems, packages and Basic software layout as follows,

(a). SYSTEMS

ADEPT:
- ADEPT
- ADEPT-SIMULATE
- ADEPT-KB
- ADEPT-SIM-KB

KATE:
- KATE
- KATE-SIMULATE
- KATE-INTERFACE

FIES:
- FIES-INTERFACE

(b). PACKAGE

USER
KATE
KATE-SIMULATE
KATE-INTERFACE

(c). BASIC SOFTWARE LAYOUT:

ADEPT
- TEST-CONFIGURATION
- DOWNLOAD
  - XMIT-RELAY-CONFIGURATION
- REFRESH

```
    CLEAR-STATUS-WINDOW-DISPLAY
RETREIVE
    RETRIEVE-CONFIGURATION
ARCHIVE
    ARCHIVE-CONFIGURATION
S-STATE
    START-FIES-STEADY-STATE-DATA-DIPLAY
B-BOARD
    START-FIES-FAULT-DATA-DISPLAY
SCEN-ARC
SCEN-RET
QUIT
    CLEAR-STATUS-WINDOW-DISPLAY
    DEEXPOSE WINDOW
    DEACTIVATE WINDOW
    KILL FIES PROCESSOR

MSFC-RULE-SYSTEM
    DIRECT-SHUNT
    RESISTIVE-SHUNT
    OPEN-RELAY
    REPORT-IT

ADEPT-SIMNULATE-CONFIGRUATION
    DOWNLOAD
        OPEN ADEPT WINDOW : ADEPT-MAIN-LOOP
        OPEN ADEPT CONTROL WINDOW :
                    ADEPT-SIMULATE-CONTROL
                    ADEPT-SIMULATION
                    EXIT-ADEPT
                    CLEAR-WINDOW
                    KILL ADEPT PROCESSOR

    REFRESH
    RETRIEVE
    ARCHIVE
    ADEPT-SIMULATE-DATA
        ADEPT-SIMULATE-DATA-DISPLAY
    SCEN-ARC
    SCEN-RET
    QUIT
        CLEAR FIES II WINDOW
        KILL FIES II PROCESSOR
```

```
;;; -*- Mode: LISP; Package: FS; Base: 10 -*-

(set-logical-pathname-host
 "Adept"
 :translations '(("adept:kate;*.*.*" "C:>adept>kate>*.*.*")
         ("adept:kate;simulate;*.*.*" "C:>adept>kate>simulate>*.*.*")
         ("adept:kate;remote;*.*.*" "C:>adept>kate>remote>*.*.*")
         ("adept:adept;*.*.*" "C:>adept>*.*.*")
         ("adept:kate;test;*.*.*" "C:>adept>kate>test>*.*.*")
         ("adept:kate;test;simulate;*.*.*" "C:>adept>kate>test>simulate>*.*.*")
         ("ADEPT:CAROLINE;FIES;*.*.*"     "C:>adept>FIES>*.*.*" )))
```

```
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: USER; Base: 10 -*-
;;; Created 6/20/86 14:15:00 by EDNEW/KSC
;;; modified 2/11/87 11:21:18 by Caroline Wang/MSFC


(defpackage kate-interface
        (:use scl)
  (:nicknames "KI")
  (:export setup-window point line circle arc square rectangle lprint))

(defpackage kate-simulate
        (:use cl)
  (:nicknames "KS"))

(defpackage kate
        (:use cl ki)
  (:import-from ki setup-window point line circle arc square rectangle lprint))

;;; definition of the KATE Simulation system on the SYMBOLICS

(defsystem kate-simulate
  (:name "Knowledge-Based Automatic Test Equipment Simulation")
  (:package "KATE-SIMULATE")
  (:pathname-default "adept: kate; simulate;")
  (:module utility "frl-util")
  (:module constraint-sys "ktester")
  (:module data-base ("kate-top"
                "kate-mid"))
  (:compile-load utility)
  (:compile-load constraint-sys)
  (:compile-load data-base))

;;; definition of the KATE system on the SYMBOLICS

(defsystem kate-interface
  (:name "KATE User and Hardware Interface")
  (:package "KATE-INTERFACE")
  (:pathname-default "adept: kate;")
  (:module comm "comm-hardware" )
  (:module main ("window-interface-vars"
            "window-interface"
            "graphics-primitives"
            "graphics-interface"
            "printer-interface"))
  (:compile-load comm)
  (:compile-load main))


(defsystem kate
  (:name "Knowledge-Based Automatic Test Equipment")
  (:package "KATE")
  (:pathname-default "adept: kate;")
  (:module utility ("frl-util"
                "symbolics-specific"))
  (:module constraint-sys "ktester")
```

```
 (:module main ("kdetect"
               "knoser"
               "nosepend"
               "k-goal"
               "maintain"
              ;"k-init"
               "kprocess"))
; (:module top "top-interface")
 (:module remote "hardware-interface")
 (:module training "test-operator")
 (:module data-base ("kate-top"
                     "kate-mid"))
 (:module graphics "draw-purge-system")
 (:compile-load utility)
 (:compile-load constraint-sys)
 (:compile-load main)
; (:compile-load top)
 (:compile-load data-base)
 (:component-systems ki::kate-interface ks::kate-simulate)
 (:compile-load remote)
 (:compile-load training)
 (:compile-load graphics))

(defsystem Adept-kb
 (:name "Automatic Detection of Electrical Power Troubles  KB")
 (:package "KATE")
 (:pathname-default "adept:kate;test;")
 (:module adept-main "main-kb")
 (:module adept-sub  ("curr-kb"
                     "currm-kb"
                     "volt-kb"
                     "voltm-kb"
                     "relay-kb"
                     "load-kb"))
 (:compile-load (adept-main adept-sub)))


(defsystem Adept-sim-kb
 (:name "Automatic Detection of Electrical Power Troubles simulation KB")
 (:package "ks")
 (:pathname-default "adept:kate;test;simulate;")
 (:module adept-main "main-kb")
 (:module adept-sub  ("curr-kb"
                     "currm-kb"
                     "volt-kb"
                     "voltm-kb"
                     "relay-kb"
                     "load-kb"))
 (:compile-load (adept-main adept-sub)))


(defsystem Adept
 (:name "Automatic Detection of Electrical Power Troubles")
 (:package "KATE")
 (:pathname-default "adept:adept;")
```

```
 (:module adept-patch ("report-it.lisp"
              "adept-init.lisp"
              "adept-main-loop.lisp"
              "adept-simulate-control.lisp"
              ))
 (:component-systems Kate Adept-kb ks::Adept-sim-kb ki::kate-interface USER::CK-FIE
S-TEST user::adept-simulate)
 (:compile-load adept-patch))
;
; (IN-PACKAGE 'KATE)
;
(defsystem adept-simulate
 (:name "adept-simulate")
 (:package "user")
 (:pathname-default "adept:adept;")
 (:module fies-kate ("adept.lisp"
              "adept-simulate.lisp"
              "read-fies-relay-configuration.lisp"
              "adept-simulate-data-display.lisp"
              "read-adept-simulate-data-table.lisp"
              "adept-exit.lisp"))
 (:compile-load fies-kate))

;;;
;;; System FIES adapted for system FIES.
;;; SWD 2/26/87

(Defsystem ck-fies-test
 (:name "ck-fies-test")
 ( :PACKAGE "USER")
 (:pathname-default "c:>adept>fies>")
;
;
 (:module defs "fies-interface-defs.lisp")
 (:module includes ("fies-get-edges.lisp"
             "fies-clear-status-window-display.lisp"
             "fies-display-status-message.lisp"
             "fies-get-configuration-file-name.lisp"
             "fies-make-configuration-file-name.lisp"
             "fies-get-scenario-file-name.lisp"
             "fies-make-scenario-file-name.lisp"
             "fies-undraw-selected-configuration.lisp"
             "fies-draw-selected-configuration.lisp"
             "fies-xmit-relay-configuration.lisp"
             "fies-power-distribution-lines-funcs.lisp"
             "fies-undraw-power-distribution-lines.lisp"
             "fies-redraw-power-distribution-lines.lisp"
             "fies-reset-power-distribution-lines.lisp"
             "fies-archive-configuration.lisp"
             "fies-retrieve-configuration.lisp"
             "fies-archive-scenario.lisp"
             "fies-retrieve-scenario.lisp"
             "fies-xmit-request.lisp"
             "fies-parse-raw-data-for-scenario.lisp"
             "fies-send-lisp.lisp"))
```

```
 (:module mouse "fies-initialize-mouse.lisp")
 (:module test-configuration "fies-test-configuration.lisp")
 (:module initial-configuration "fies-initial-configuration.lisp")
 (:module serial-interface "fies-serial-interface.lisp")
 (:module serial-interface-test-refresh "fies-refresh-data-display.lisp")
 (:module steady-state-data-display "fies-steady-state-data-display.lisp")
 (:module fault-data-display "fies-fault-data-display.lisp")
; (:module fault-isolate "fies-fault-isolate.lisp")
 (:module serial-interface-test "fies-serial-interface-test.lisp")
 (:module msfc-rule "ck-fies-rule-test.lisp")
;
;
;
 (:compile-load defs)
 (:compile-load includes (:fasload defs))
 (:compile-load mouse (:fasload defs))
 (:compile-load serial-interface-test-refresh (:fasload defs includes))
 (:compile-load steady-state-data-display (:fasload defs includes))
 (:compile-load fault-data-display (:fasload defs includes))
; (:compile-load fault-isolate (:fasload defs includes))
 (:compile-load initial-configuration (:fasload defs includes mouse))
 (:compile-load serial-interface (:fasload defs includes))
 (:compile-load serial-interface-test (:fasload defs includes))
 (:compile-load test-configuration (:fasload defs includes mouse))
 (:compile-load msfc-rule))

(IN-PACKAGE 'KATE)
```

2. Software Function Descriptions

The following pages give the software function descriptions.

```
FRL-UTIL                            : The frame Base Inference Engine Utility

  (Frame? frame)                    : Display the contents of the frame
  (mframe frame)                    : Display the contents of the frame
  (Deframe name & rest slots)       : Define new frame name and slots
  (Fremove frame slot &optional value)
                                    : Remove the slot from a frame
  (Fremove-val frame slot value)    : Remove value from slot
  (Fremove-slot frame slot)         : Remove slot from a frame
  (Fput frame slot value)           : Replace the new value to the slot
  (Fadd frame slot value)           : Add new value to the slot
  (Extend key a-list)               : display the slot value of the frame
                                      (extend 'cvalue (mframe 'c1))
  (Mget frame slot)                 : Get the slot value
  (Cmget frame slot)                : The CAR of mget
  (Mgetl frame slot)                : Get the slot value
  (Cmgetl frame slot)               : The CAR of mgetl
  (Mget-I frames slot)              : The slot value of a list of frames
  (Mget-Frames frame)               : Backward channing the instance of the frame
  (F-replace frame slot newval)     : Replace new value of the new slot within the
                                      frame
  (Freplace frame slot newval)      : Same as F-replace
  (Fremove-slot-values frame slot): Remove slot value from the frame
  (Mklist value)                    : Make a list
  (Newpush item list)               : Push a new item into the list
  (Ako? fr gen &optional (colors '(aio ako)))
                                    : A kind of
  (/*fvalues-only frame slot)       : Macro for get the slot value
  (subset fn fnlist)                : A new list is made by applying
                                      fn (a predicate) to all the elements of
                                      fnlist and removing the ones for which the
                                      function return nil.
  (Subset-not fn fnlist)            : The opposite of subset
  (Ldifference a b &aux new-list) : List difference. returns a list of those
                                      elements in a that are not memvers of b
  (Flatlist x)                      : a list
  (First-Image somex somefnl)       : The solution of the funcall for the list of
                                      items
  (Sym-some somelist pred)          : Same as First-Image
  (Sym-listp x)                     : If x is a list
  (Sym-nlistp x)                    : If x is not a list
  (Intersect &rest lists &aux result)
                                    : Intersection
  (Adjoin-equal item list)          : If item is a member of list, then list the
                                      list.
                                      If item is not a member of list, then join
                                      the list.


KTESTER                             : (Kate simulate Package)

  (Simulate-Fail object value)      : Replace the new value for the simulated objects
  (Excute-Control-Function command-frame value)
                                    : Search for the Command-frame in Kate Simulate
                                      Package and replace the new value to it.
  (Sim-Val frame)                   : Simulate the current measurement of the Frame
                                      and compare it with the measurement in the
                                      Queuearray
  (Simulate-Measurement measurement-frame)
                                    : Simulate measurement and add the current time
                                      to the list
  (Statnoval object &aux val)       : Get the current status value from the object
  (Evalget object slot)             : Get current slot value of the object
  (Statval object)                  : Get current slot value and put on status list
```

```
(Do-statval object &aux val analog?)
                                : Get current slot value
(Cstatus obj-expr)              : Get current status (Macro code) and put it
                                  on status list
(Cstatus1 obj-expr)             : Get current status (Macro code)
(Ad-cstatus expr)               : A to D Current status , Backward Chainning
                                  and searching for related status.
(Ad-statfn exp)                 : Read A to D current status
(Check-Downstream frame)        : Check the object downstream activity
(Analogp object)                : If it is an analog object
(Discretep object)              : If it is a discrete object
(Power-off-status)              : Get power off status
(Boolify discrete-value)        : If discrete-value is 'on or t, then return t.
                                  If discrete-value is 'off or nil, then return
                                  nil. Otherwise return discrete-value
(Unboolify boolean)             : If discrete-value is 'on or t, return to 'on.
                                  If discrete-value is 'off or nil, return 'off.
                                  Otherwise return boolean.
(Add-to-pending-list frame expiration-time)
                                : Add the new object and expiration time to the
                                  pending list.

(Remove-From-Pending-List frame &aux (point (member frame *pending-list*)))
                                : Remove the object from pending list.
(Do-pending-List)               : Execute the object when time expired and then
                                  remove it from pending list and check
                                  downstream.
(Do-Pending-List1)              : Execute the object when expired and remove it
                                  from pending list.
```

KDETECT

```
(Detects-P po pov fo fov suspects &aux alt-pos alt-com ...)
                                : Dectect the current state.
(Common-Controlling-Objects f1 f2)
                                : Objects checkd for controllers
(Controllees commands objects)  : Controller link between object and command.
(Immediate-Controllers object &aux objs)
                                : Replace Immediate controller object
    and push it to the *objects-check-for-controllers* list.
 (Stat-replace expr var new)      : This substitues (via string manipulation)
                                  new for either (cstatus var) or
                                  (statval (quote var)) in expr wherever they
                                  occur.
(Repx expr old new &aux found)   : This substitutes (via string manipulation)
                                  new for old in expr if old occurs. The expr
                                  is returned or nil if no matches are found.
(Lower-Bound frame)              : get lower-bound of the object or 0.0
(Uper-Bound frame)               : Get uper-bound of the object or 100.0
```

KNOSER : KATE DIAGNOSER

```
(Iball fo fov status *aux siblings ...)
                                 : Search for the failure and prints the report.
(do-suspects fo fov siblings suspects &aux ...)
                                 : Update display siblings, Print consistency
                                   rational suspect.
(Get-siblings suspects fo &aux...): Downstream measurements suspect failure
                                   object to suspect-siblings.
(Find-one-sibling suspects discrete-power-sibling &aux...)
(Downstream-measurements ancestors object &aux measurements)
                                 : search for ancestors measurements.
```

```
(Print-items items)                : Print item nomenclature, cvlue...
(Print-siblings&suspects siblings suspects)
                                   : Print report.
(Sort-siblings siblings object &aux...) Sort sibling list
(Print-anticipated-result suspects) Print the failure
(Print-ipr-conclusions culprit)     : Conclusion
(Print-remaining-suspects suspects) : Conclusion Remaining suspects
(Print-screwed)                     : Apology for no single point failure
                                      possible for this situation.
(Get-Time-Date)
```

K-GOAL

This is an attempt to design a new goal attaining function using a different approach.

```
(Goal object value &optional just-do-it)
                                    : This is a function to achieve the given
                                      object value. Goal use attain to give
                                      requirements for what needs to be done to
                                      achieve the target and then follow the
                                      procedure to do it.
(Attain object value)               : This function gives the requirements for
                                      what needs to be done to achieve the new
                                      object value.
(attain-discrete object value &aux temp)
                                    : If the object is discrete object then use
                                      attain  discrete.
(Follow-Logic source-path value &aux logic-list)
                                    : Follow the Logic backward chainning the
                                      object value through source-path.
(Attain-analog object value &aux temp)
                                    : If the object is an analog object, then
                                      use attain-analog.
(Analog-follow-Logic object value source-path power-off &optional notted &aux
    logic-list
 temp-alt)                          : Follow the Logic backward chainning the
                                      analog object value.
(Mid-Pair value) :
(First-Pass options &aux logic-list)
                                    : Push first pass option to the logic list.
(Remove-already phrase) :
(In-path-of? command)               : Finds the objects in a command's
                                      "in-path-of" slot.
(Sinks? command)                    : Finds the objects in a command's "sinks"
                                      slot.
(Check-for-maintain command)        : Check object for maintaining
(Check-for-Maintain1 object command)
                                    : "bail-out" if object is found that is
                                      being maintaining.
(Option-not-okp object)             : The object can not be maintained.
(Second-pass option just-do-it &aux good-ones good-cmnds)
                                    : Append the good choice command to the
                                      good-ones list
(Good-Choice? commands &aux old-list)
                                    : Skip the failed command and save the Good
                                      choice command.
(Restore-values old-list)           : Replace the new value to the old-list.
(And-Or tree)                       : If (CAR tree) is 'and, then combine elements
                                      If (CAR tree) is 'or, then chose either
                                      element.
(Do-And tree &aux trimming)         : Combine elements
(Order-Decreasing options &aux new-list)
                                    : Remove first option and make a new option
                                      list.
(And-combine-elements options &aux final)
```

	: Combine the elements into a list
(Display-alts)	: Displays all of the "minimally sufficient" alternatives to reaching the goal.
(Menu1 &aux opt)	: Menu for user input of option number to be used to accomplish goal and allows for viewing of those options that were weeded out.
(Rounder number decimal-places)	: Set decimal-places equal or less than 5.

MAINTAIN

File for the maintaince functions needed to impliment automatic redundancy managemenet

(Clena-up)	: Reset
(Maintain object value)	: If the object is a frame, then use goal function to achieve the value for the object.
(Maintainedp object)	: Check if the object is a maintained object.
(Unmaintain object &optional state)	: Remove the object from the maintained list.
(Unmaintain-all)	: Remove all the objects from the maintained list.
(*Maintained*)	: Property list for maintained objects.
(Fail object state)	: Put the failed object state to the failed list.
(Unfail object)	: Remove the object from the failed property list.
(Suspect object state)	: Put the suspect object into the suspect object list.
(Unsuspect object)	: Remove the object from the suspect object list.
(Unsuspect-all)	: Remove all the objects from the suspect object list.
(*Failed*)	: Property list *failed*
(*Suspect*)	: Symbol property list *suspect*
(Failedp object)	: If it is a failed object.
(Nfailedp object)	: If the object is not a failed object.
(Suspectp object)	: If it is a suspected object.
(Check-for-maintain-no-fail object &aux object-found)	: Check the object is a maintained object.
(Failed-maintain object)	: The object is not in the maintained object list.
(Unfail-maintain object)	: list the maintained object.
(Unfail-all)	: List all object for maintained list.
(Recover object &aux options)	: Look for fail-maintain list and maintain the object.
(Get-Well suspects &aux can-do)	: Recover all suspect items which are on the fail-maintain list.

KPROCESS

KATE process procedure

(Run-procedure-step)	: Checks the *delay-step* and *pending-list* to process KATE.
(Process-queue)	: Takes the next item on the *queuearray*, updates *Kate-pointer * and hand the function designator and value to insert-fd.
(Insert-fd fd value)	: Updates fd's current value slot and if it is a command, then runs check-downstream, otherwise runs test-val.

(Timer)	: Current computer time.
(Future-Time delay)	: Current time plus (60*delay).
(Sec-Time atime)	: (60*atime) ;;;atime and delay are both in seconds.

REPORT-IT

(Report-it) and (Write-Kline) are both modified function for sending messages to lisp-listener other then KATE window.

ADEPT-INIT

(Initialize-Adept)	: Set up arrays, pointers and lists for ADEPT simulater.

ADEPT-MAIN-LOOP

(Adept-main-loop)	: Running the KATE processor main loop.

ADEPT-SIMULATE-CONTROL

(Adept-simulate-control)	: This function creates a momentary-menu window to run Adept-Simulation.

ADEPT-SIMULATE

(Adept)	: Top-level window, display options to select 1. Run Adept real time fault Isolation 2. Run Adept Local Simulate or 3. to Exit Adept.
(Read-Fies-Relay)	: Read the Relay configuration through Fies II window interface and send the goal to Kate inference engine. After the goal achieved, th e Knowledge Base will be updated.
(Adept-Simulate-Configuration)	: Run Adept simulation through FIES II interface.
(Adept-Simulate-Data-display)	: Display Adept Simulation data on FIES II window.
(Read-Adept-simulate-data-table)	: Read Adept Simulation data table from array.
(Adept-exit)	: Exit Adept Simulater, clear all Adept windows and kill Adept processor.
(CK-Fies-Rule-test)	: MSFC Adept Generic rule system.

CK-FIES-TEST

(Fies-interface-defs)	: FIES variable declarations and initializations.
(Get-Edges)	: Get the main-screen edges.
(Fies-clear-status-window-display)	: Clear fies status array.
(Fies-display-status-message)	: Display current status message in the message window.
(Get-configuration-file-name)	: Input file name for retrieve new configuration.
(e-Scenario-file-name)	: Input file name and open a file for scenario file.
(Undraw-selected-configuration)	: Undraw selected saved configuration.
(Draw-selected-configuration)	: Redraw selected configuration.
(Xmit-Relay-configuration)	: Transfer relay configuration through serial 3 to Intel Micro Processor.
(Power-distribution-lines-funcs)	: Draw path, Undraw path and Calculate Relay status Index.
(Undraw-Power-distribution-lines)	: Uedraw lines from relays to load modules.
(Redraw-power-distribution-lines)	: Redraw lines from relays to load modules.
(Reset-Power-distribution-lines)	: Reset lines from relays to load modules.

(Archive-configuration)	: Archive configuration from saved file.
(Retrieve-configuration)	: Retrieve configuration from saved file.
(Archive-scenario)	: Put Archived configuration to open file
(Retrieve-scenario)	: Put Retrieved configuration to open file.
(Xmit-Request)	: Download an transmit buffer to the micro processor.
(Parse-raw-data-for-scenario)	: Parse io buffer into table.
(Send-lispt)	: Not used for Adept.

KNOWLEDGE BASE

KATE-TOP, KATE-MID	
MAIN-KB	: ADEPT Top level knowledge base.
CURR-KB	: Current knowledge base.
CURRM-KB	: Current measurements.
VOLT-KB	: Voltage Knowledge base.
VOLTM-KB	: Voltage measurements.
RELAY-KB	: Relay knowledge Base.
LOAD-KB	: Critical Power Loads, Low Power Loads and High power loads Knowledge base.

3. SDAS Intel Micro Processor Interface Flow Diagram

Figures 4 and 5 are flow diagrams of the SDAS Intel Microprocessor Interface.

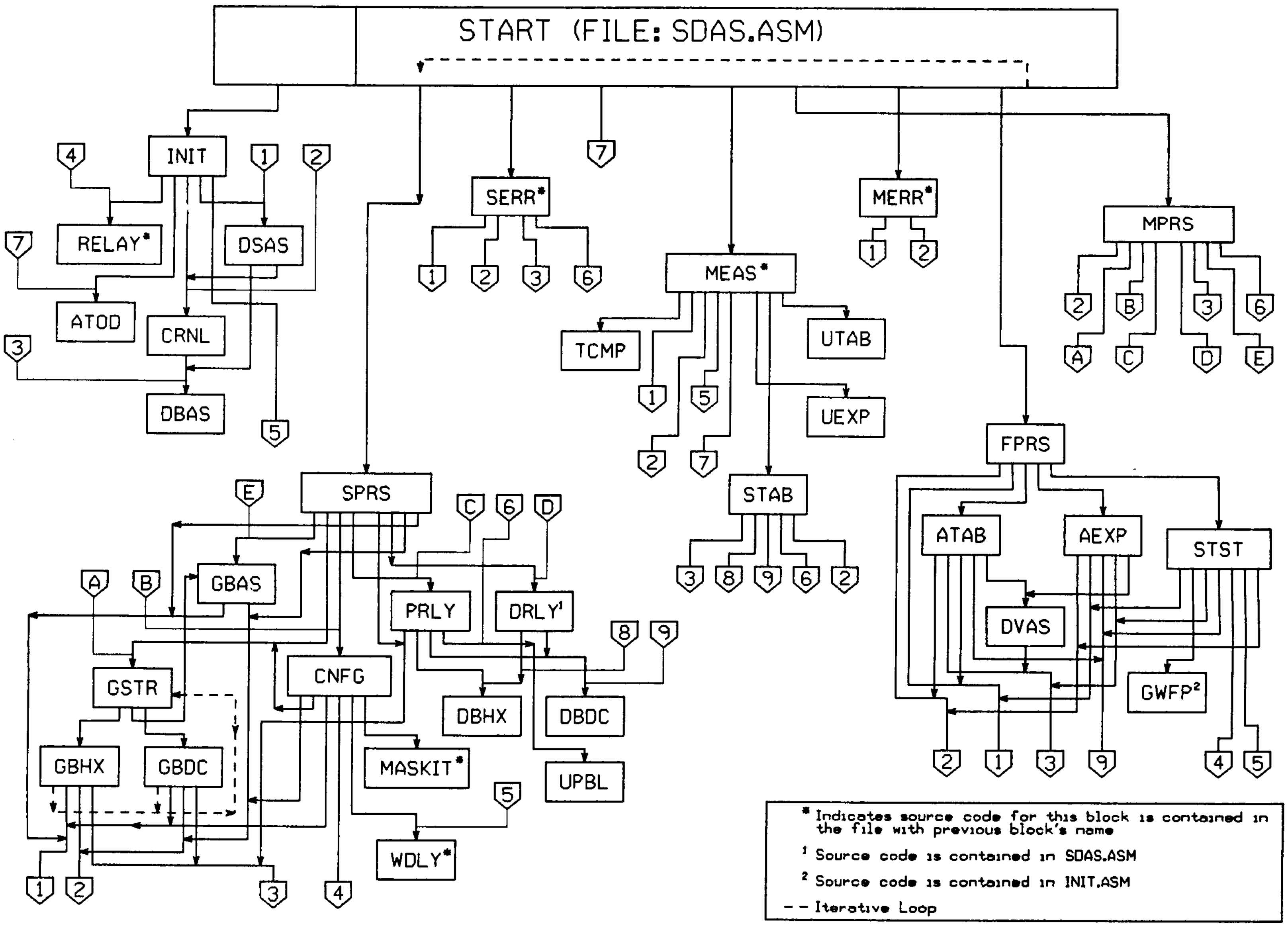


Figure 4. SDAS Subroutine Flowchart.

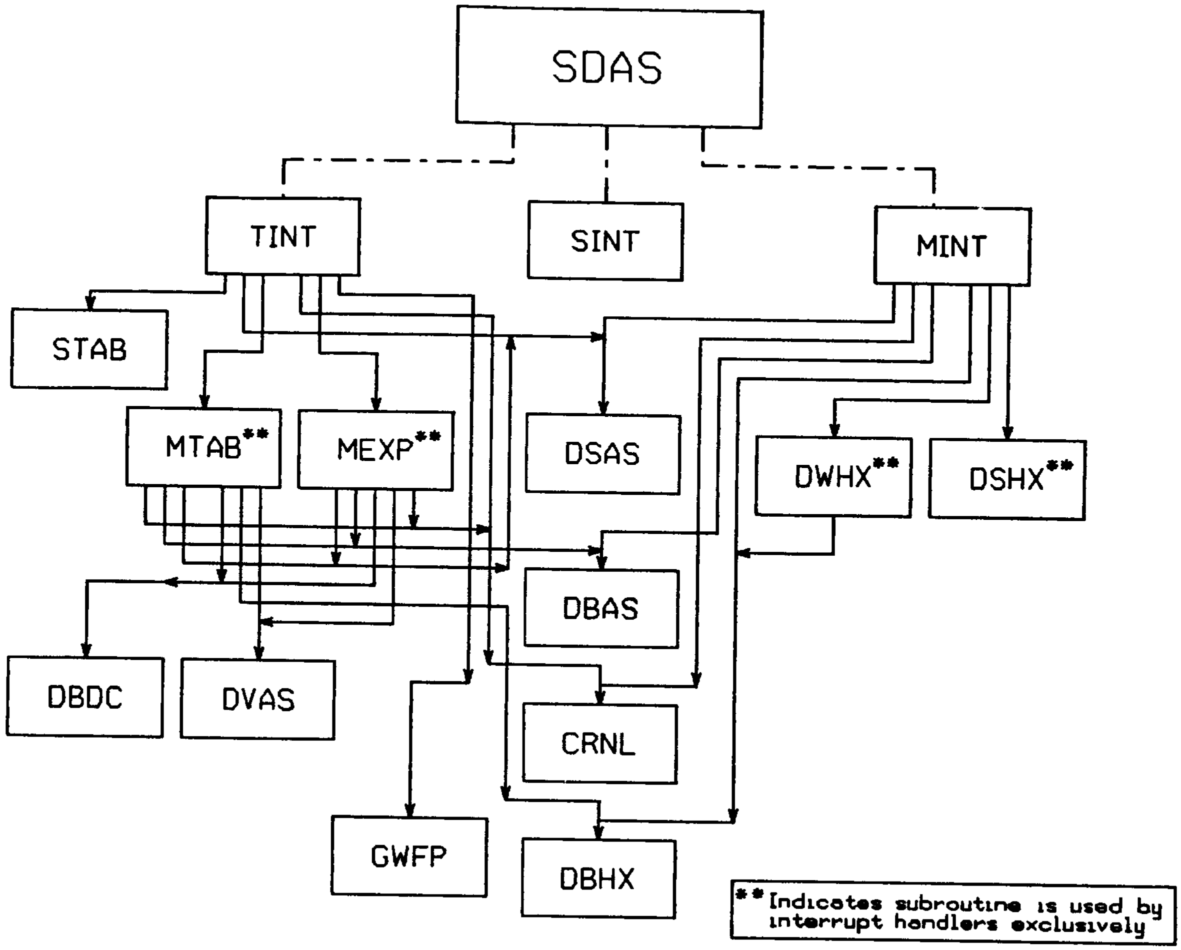


Figure 5. SDAS Software Interrupt Handler Flowchart.

4. SDAS Program Subroutine Descriptions

The following pages give a description of the SDAS program subroutine.

SDAS.EXE is the command which begins execution of the Sensor Data Aquisition Scheduler (SDAS), software responsible for control of both the FIES II breadboard and its interaction with related hardware. The SDAS program exists as forty-one ASM86 Assembly Language modules in thirty files on the Intel System 86/380 computer built into the FIES II hardware rack. Further information on the 86/380 is contained in the hardware summary pp. 37-38

Flow of program control among the forty-one modules is diagrammed in [figures 4-4(a) and 4-4(b)]. Conventions used in these charts are as follow.

Boxes in the chart contain the names of the forty-one modules, and unless footnoted otherwise, these names are the same as the names of the files in which the source code is contained. Heavy lines extending from the bottoms of the boxes connect them to subroutine modules, either directly or through numbered reference tabs. Note that, in general, original flow is downward and returns are serviced upward through the chart; arrowheads are for reference only, and should clarify any exceptions or points of possible confusion. Also note that the lines used to connect the numbered tabs to their respective reentry points are lightened for simplicity. Finally, many of the modules listed in the second figure can be found originally in the first; these were listed again for simplicity, and those modules unique to the interrupt handlers are marked as such.

SDAS software can be divided into three sets: the initialization handler, normal run-mode routines, and the interrupt handlers. Once the second of these is accessed, program control cycles through the normal routines continuously, as shown by the dashed arrow in the START module box, until normal execution is interrupted. Different sections of this cycle are dedicated to the control of input and output to and from the Symbolics, debug monitor, and the front panel of the FIES II hardware rack.

Each of a module's exiting lines can represent any number of calls made by the module to the subroutine to which that line connects. The order in which the subroutines are first called by a module can be read in the left-to-right arrangement of the exiting lines; the number of calls made to each subroutine and the order of subsequent calls are left out for simplicity.

The flow diagrams should be used to gain a general understanding of the interactions of the modules contained in the SDAS. To further this end, a list is provided below of the different modules, the meaning of their abbreviations where applicable, and a short statement of the function of each. Module names appear roughly in the order of main routines first followed by lower subroutines as they would be encountered during program execution. The latter are divided into groups that pertain to the Symbolics, monitor, or front panel controller sections of the flow diagram. The last grouping is for those modules contained in the interrupt handlers.

START	Main controller for the software package
INIT	Initialization Handler: sets variables, interrupt vectors, board addresses, pointers, etc., prior to main program execution
SPRS	Symbolics command parser
SERR	Symbolics error reception handler (halts execution)
ATOD	Analog to digital measurement processor
MEAS	Measurement controller: handles timing of calls to the A/D
MPRS	Monitor command parser
MERR	Monitor error reception handler (halts execution)
FPRS	Front panel command parser
RELAY	Controls relay states through wire-wrap board in Intel rack
CRNL	Carriage Return/New Line sequence output
DSAS	Display string of ASCII (see DBAS)
DBAS	Display byte of ASCII
GBAS	Get byte of ASCII (from monitor or Symbolics)
GSTR	Get string of characters
GBHX	Get byte of hex data from two ASCII-encoded hex characters
GBDC	Get byte of ASCII-encoded decimal
CNFG	Configuration manager: downloads power system configurations from the Symbolics or the monitor
WDLY	Delay timer for INIT, CNFG, and STST
MASKIT	Masks relay control bits
PRLY	Displays the state of a specified relay
UPBL	UnPadded blank (EOL) character output
DRLY	Displays states of sixteen relays (one mask word)
DBHX	Display byte of ASCII-encoded hex
DBDC	Display byte of ASCII-encoded decimal
TCMP	Table data comparator: compares previous with present A/D readings
UEXP	Updates exception table data
STAB	Sends raw measurement data table to Symbolics
UTAB	Updates measurement table data
ATAB	Displays measurement table data on ASCII display device
AEXP	Displays exception data on the ASCII display device
DVAS	Display measurement values
STST	Self test sequencing mode, accessed through front panel switch
GWFP	Get control word from front panel
TINT	Terminal interrupt handler
MTAB	Displays raw measurement data at monitor
MEXP	Displays exception data at monitor
SINT	Symbolics interrupt handler
MINT	Monitor interrupt handler
DWHX	Display word of ASCII-encoded hex
DSHX	Display string of ASCII-encoded hex

C. Basic Hardware Layout

1. Hardware Flow Diagram

Figure 6 is a flow diagram of the SDAS hardware.

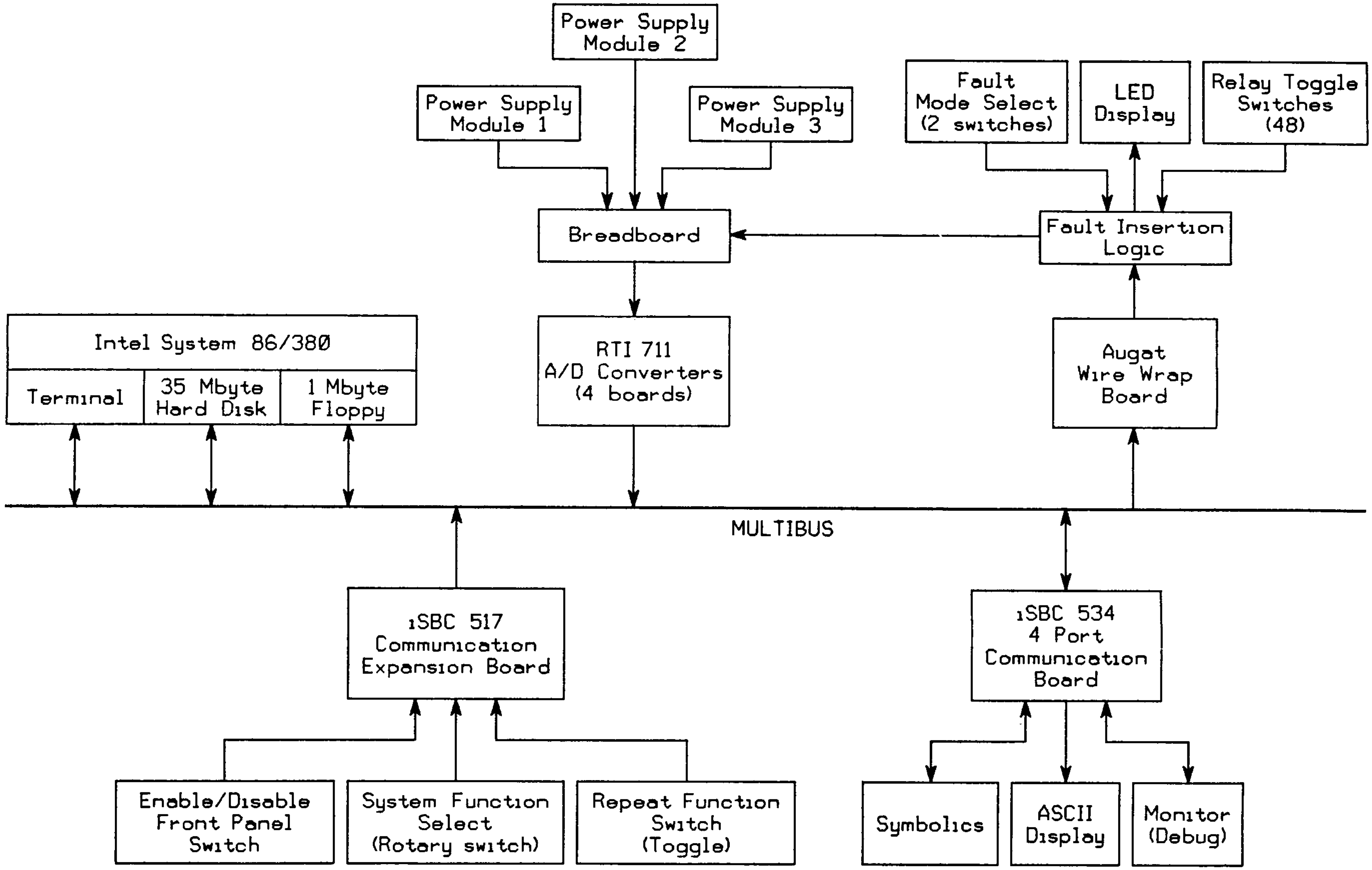


Figure 6. Hardware Flow Diagram.

2. Power System Breadboard

The following pages show the power system breadboard.

Part of FIES II is built into two side-by-side racks containing the host computer with memory storage devices and I/O support equipment; the relay board subrack, its power supplies and related controllers; communications boards, ports, and cables; housekeeping power supplies; control switches and lighted displays, including the ASCII LED display; and all of the connectors, wiring and power strips necessary to the function of the system. The only FIES II hardware components not built into these racks are the Symbolics computer used for the artificial intelligence applications portion of the system and, of course, the freestanding CRT displays. That portion of the system contained in the racks is often referred to as "the breadboard," as will be hereafter.

Refer now to figure 5-5 for an outline of the components of the FIES II system and the interactions of these components with one another.

Data transfer scheduling and control are provided by the host computer, an Intel System 86/380. Based on the iRMX86 operating system, the 86/380 contains the iSBC 86/30 Single Board Computer board, a thirty-five megabyte Wincester hard-disk, a one megabyte eight-inch flexible disk drive, and a multibus expansion rack with slots containing not only controllers for the computer itself, but also certain other boards discussed below. Software run on the 86/380 for the FIES is written in Intel's ASM86 assembly language; this program, SDAS, was discussed in the previous section.

The heart of the breadboard is the relay board subrack, located in the middle of the left rack. This is comprised of six boards containing forty-eight relays along with related resistors, capacitors, and other components represented by the simple diagram of the system found in/on p. 46 , neglecting the three power supply modules. In addition to its function as the system's switching center, the relay subrack provides attach points for all but three of the sensor lines to the A/D converters and for the fault insertion lines.

Three dual-sided power supplies provide either charge to the batteries or electricity to drive the system's load resistances, or both, depending upon the configuration into which the relays are set. Simulating the Space Station's solar arrays, these supplies are capable of up to fifty volts and nearly two amperes output on each of the six available channels. Each set of two channels must be balanced so that either side will provide the same current as the other when both are connected in common; if any pair is very much out of balance, a fault may well be detected. Each supply has independent current-limiting adjustment, allowing simulation of various solar array lighting conditions.

The fault insertion logic, used to introduce various abnormal loads at nodes along the power modules' circuits, sends its outputs directly to the relay boards but can receive inputs in different ways. Manual operation of the switches on the fault insertion panel, located on the right rack toward the top, is one method. Forty-eight toggle switches place individual relays in normal, relay, or shunt modes; the "relay fault type" switch selects open or closed faults, the "shunt fault type" switch sets either resistive or direct shunt faults, and normal positioning of a toggle switch ignores the settings of both fault type switches, leaving control of that relay to the computer. Configuration of the relay switches can thus also be accomplished by downloading relay controls from the Symbolics or the monitor through the Augat wire-wrap board, which takes up one of the slots on the 86/380. In the event that a switch is offset from the normal mode, the corresponding relay cannot be controlled remotely, but a fault will exist and should be detected by the Symbolics.

A few additional points should be noted concerning the fault insertion panel. First, the fault type switches are not programmable, and must be set manually; similarly, the actual faults must be set at the panel using the relay switches. LED's on the panel light up only when the corresponding relays are actually closed, and will not light if, for instance, the relays are set to be closed but the relay fault type is "open."

Other functions accesed manually from the FIES racks are those related to the front panel, located in the left rack adjacent to the fault insertion panel. The front panel contains the main power switch, system and load branch power indicators, a rotary function select switch with a "repeat function" switch to resubmit the selected function, a thumbwheel selector for indicating one of the twenty-four sensor nodes, and an ASCII LED display for readout of node data. The rotary dial can select reset, status mode, Symbolics slave mode, a relay self-test routine, or monitor-directed test mode; other settings shown on the knob are presently not used. All of the front panel controls mentioned here are serviced by the 86/380 through an iSBC 517 Combination I/O Expander board in its front slot.

Communication of data to the ASCII display or with the monitor and Symbolics is by means of an iSBC 534 Communications Expansion Board, also found in the 86/380 Multibus slot. This board has one parallel port and four serial RS232-compatible ports; only three of the latter are used. As suggested, the ASCII display can only receive data, and usually only shows a system condition message unless it is being used to read out node data with the status function. On the other hand, both the monitor and the Symbolics are capable of full two-way communication.

Normal operating procedure is to power up the system, turn on the 86/380 terminal, set the function select switch on the front panel to RESET, enter SDAS.EXE on the terminal, and select SLAVE MODE on the front panel. In the event of unexpected results or problems, one should be sure to check that all power supplies are on, including the five volt and twelve volt housekeeping supplies, and that the three dual supplies are balanced well. Note that the twelve volt supply is inside the left rack, behind the battery modules. Also, it is occasionally necessary to reset one or more of the three breakers on the lower front of the left rack; these are tripped by excessive current in the battery circuits.

V. ADEPT USER'S GUIDE

A. ADEPT Installation Guide

The following is the ADEPT Installation Guide.

ADEPT INSTALLATION GUIDE

1. Complete copy ADEPT directory to the new system
Key in Select F
Click LMFS
Key in copy file : host:>adept>*.*.*
Copy ADEPT.system and ADEPT.translations files into
host:>sys>site>
Key in copy file from host:>adept>adept.system
to host:>sys>site>adept.system

Key in copy file from host:>adept>adept.translations
to host:>sys>site>adept.translatations

2. Make ADEPT SYSTEM
(Make-system 'ADEPT ':compile ':noconfirm)

B. ADEPT Operating Procedure

The following gives the ADEPT operating procedure.

ADEPT OPERATING PROCEDURE

1. Turn Breadboard Main Power Switch on

2. Reset Breadboard

3. Boot up Intel Microprocessor

4. Run Intel communication program by key in
   SDAS.EXE

5. Turn Breadboard control switch to self test mode
   Test the Bread board

6. Turn Breadboard control switch to slave mode
   Ready to communicate with Symbolics system

7. Run ADEPT on Symbolics 3670 by key in
   (ADEPT)

8. Display options on the screen using mouse:

   1. Real Time Fault Isolation
   2. Simulate Local
   3. Exit

9. If option 1 was selected
   then FIES II window opened

   Click the selected relay configuration
   Click the middle-mouse for pop up menu

   Display options: Download
   Refresh
   Retrieve
   Archive
   S-state
   B-state
   Scen-ret
   Scen-arc
   quit

   If option "DOWNLOAD" was selected
   The Symbolics will send the download configuration through Intel
   Micro Porcessor to the breadboard.
   Wait until the "STEADY STATE" achieved,
   and "FAULT DATA" sign displaied on Symbolics screen when
   it is ready for Fault insertion from the Fault panel.
   After the Fault is inserted, the fault will be quickly detected,

display and reasoning is also provided.

10. If simulate local was selected
then the Modified FIES II window for ADEPT simulater version will be opened.
Click the selected relay configuration
Click the middle-mouse for pop up menu

Display options: Download
Refresh
Archive
Adept-Simulate-Data
Scen-Ret
Scen-Arc
quit

If "DOWNLOAD" option was selected
then the ADEPT window and ADEPT control window will be opened.
Moused to ADEPT window and key in

(kate::adept-main-loop)
Moused to ADEPT Control Window and key in
(Adept-Simulate-Control)
There will be a pop up window
options: 1. Run ADEPT Simulation
2. Exit ADEPT

If option 1 was selected,
It will send the Relay configuration to KATE inference Engine and
update all the Simulated sensor points valtage and current values
in ADEPT knowledge Base.

The pop up window will be on again.

If option 2 was selected, then all the ADEPT windows will be cleared
and the ADEPT processor also will be killed.

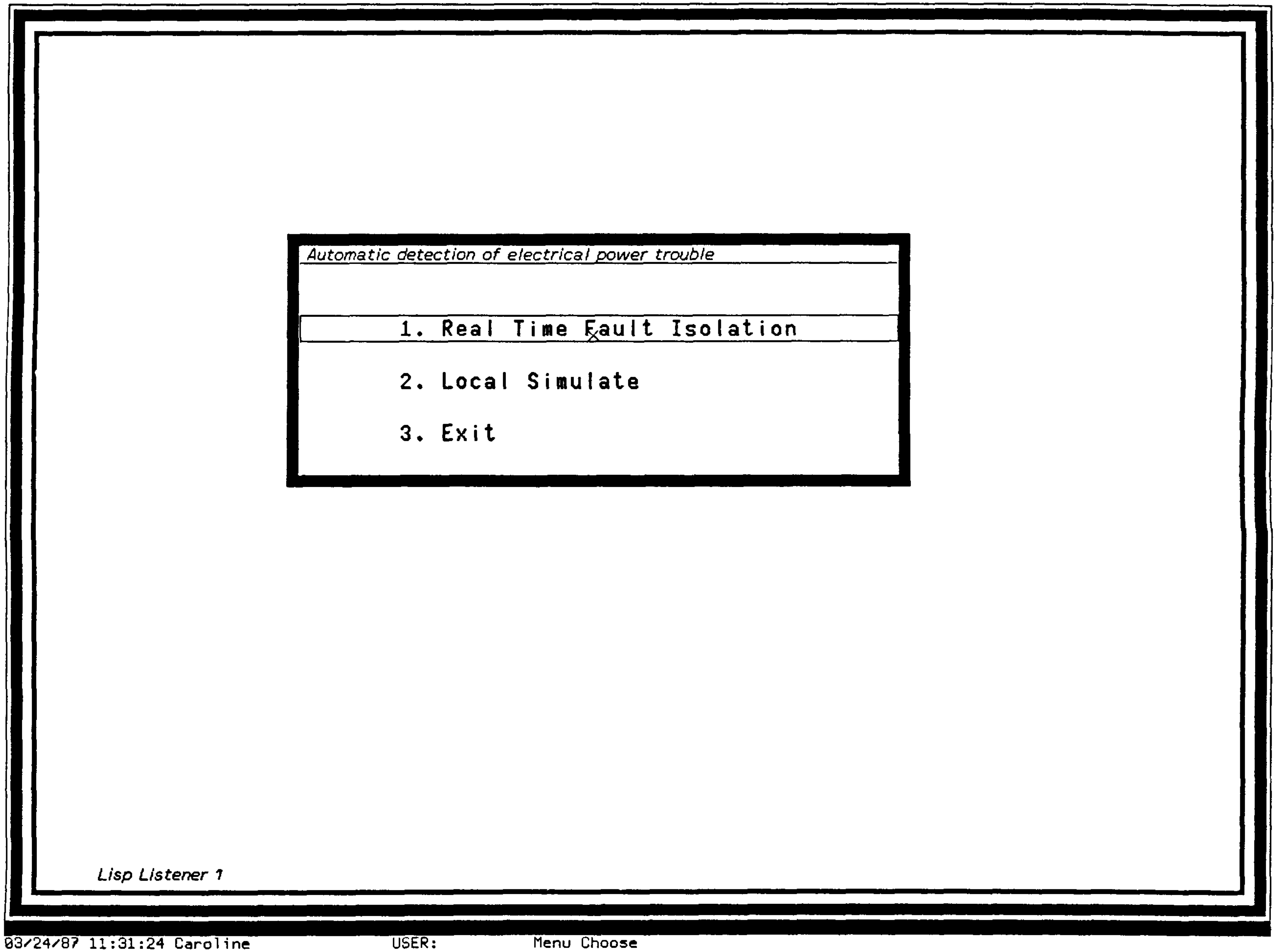
Return to the modified FIES II window, choose the ôption
"ADEPT-SIMULATE-DATA" which will display all the current simulation
data on the FIES II screen.

11. Click middle-mouse, FIES II pop up a option menu:
Select quit option to clear FIES II window and kill the FIES II interface
and processor.

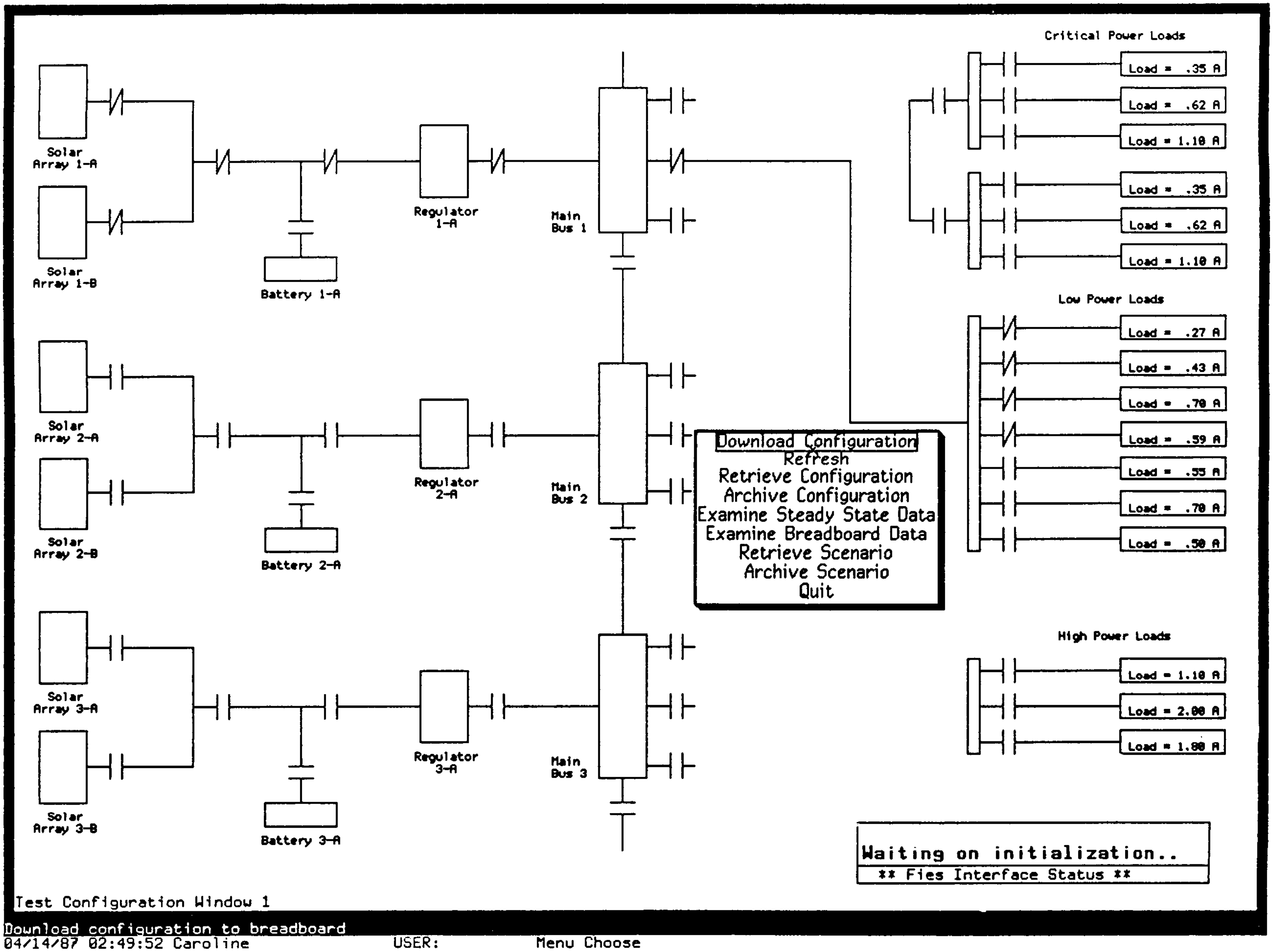
The system is back to Lisp Listenser.

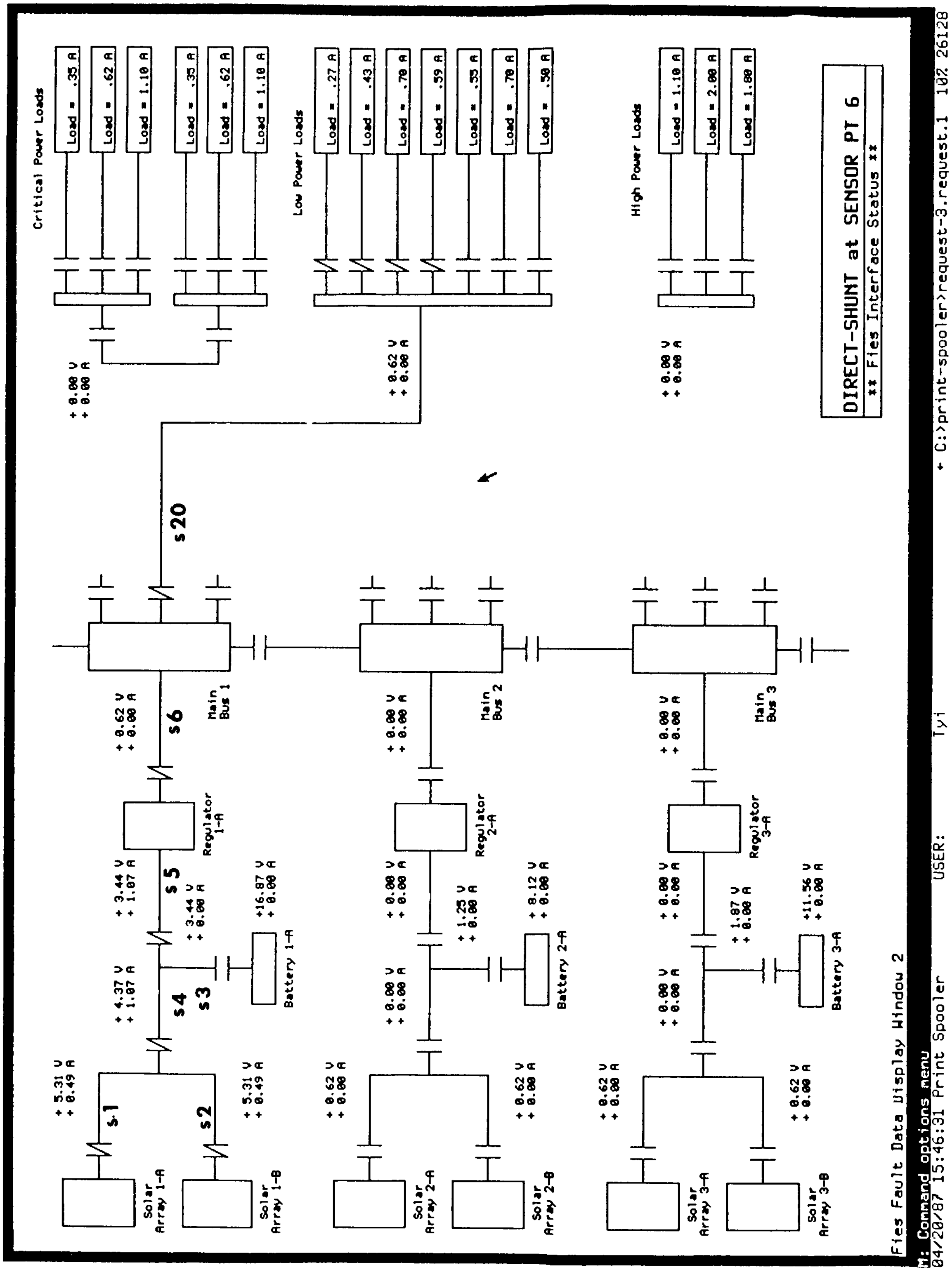
C. ADEPT Operating Example

The following gives an ADEPT operating example.


Automatic detection of electrical power trouble
1. Real Time Fault Isolation
2. Local Simulate
3. Exit
Lisp Listener 1
03/24/87 11:31:24 Caroline USER: Menu Choose

ORIGINAL PAGE IS
OF POOR QUALITY




Critical Power Loads
+ 0.00 V
+ 0.00 A
Load = .35 A
Load = .62 A
Load = 1.10 A
Load = .35 A
Load = .62 A
Load = 1.10 A
Low Power Loads
+ 0.62 V
+ 0.00 A
Load = .27 A
Load = .43 A
Load = .70 A
Load = .59 A
Load = .55 A
Load = .70 A
Load = .50 A
High Power Loads
+ 0.00 V
+ 0.00 A
Load = 1.10 A
Load = 2.00 A
Load = 1.80 A
+ 5.31 V
+ 0.49 A
s 1
Solar
Array 1-A
s 2
+ 5.31 V
+ 0.49 A
Solar
Array 1-B
+ 4.37 V
+ 1.07 A
s4
s3
+ 3.44 V
+ 1.07 A
s 5
+ 3.44 V
+ 0.00 A
Battery 1-A
+16.87 V
+ 0.00 A
Regulator
1-A
+ 0.62 V
+ 0.00 A
s6
Main
Bus 1
s20
+ 0.62 V
+ 0.00 A
Solar
Array 2-A
Solar
Array 2-B
+ 0.62 V
+ 0.00 A
+ 0.00 V
+ 0.00 A
+ 0.00 V
+ 0.00 A
+ 1.25 V
+ 0.00 A
Battery 2-A
+ 8.12 V
+ 0.00 A
Regulator
2-A
+ 0.00 V
+ 0.00 A
Main
Bus 2
+ 0.62 V
+ 0.00 A
Solar
Array 3-A
Solar
Array 3-B
+ 0.62 V
+ 0.00 A
+ 0.00 V
+ 0.00 A
+ 0.00 V
+ 0.00 A
+ 1.87 V
+ 0.00 A
Battery 3-A
+11.56 V
+ 0.00 A
Regulator
3-A
+ 0.00 V
+ 0.00 A
Main
Bus 3
DIRECT-SHUNT at SENSOR PT 6
** Fies Interface Status **
Fies Fault Data Display Window 2
M: Command options menu
04/20/87 15:46:31 Print Spooler USER: Tyi
← C:>print-spooler>request-3.request.1 10% 26128

POWERSYSTEM FAULT ISOLATION EXPERT SYSTEM REPORT

Fault Type :Direct Shunt

Reasons:::

****

Direct shunt fault casuses a sudden increase in the sensor readings of current values and a decrease in voltage values on sensors nearest the power source. When this occurs the type fault is identified and a search begins for a sensor point where the current reading is nearly zero. The fault is located between the sensor where the current is higher than the steady state current and the one where the current reading is approximately zero.

****

REASON FOR DIRECT SHUNT:
At relay point 1,
The Fault voltage 5.312486 is less than half of the Steady State voltage 19.687449

LOCATION:
DIRECT SHUNT AT Relay point 6 Sensor point 6,
The fault current 0.0 is either 0.0 or less than 0.0 or
the voltage 0.6249984 is less than 1.0

OUTPUT TABLE : R : Relay Point
S : Sensor Point
S.V. : Steady data Voltage
F.V. : Fault data Voltage
S.C. : Steady data Current
F.V. : Fault data Current

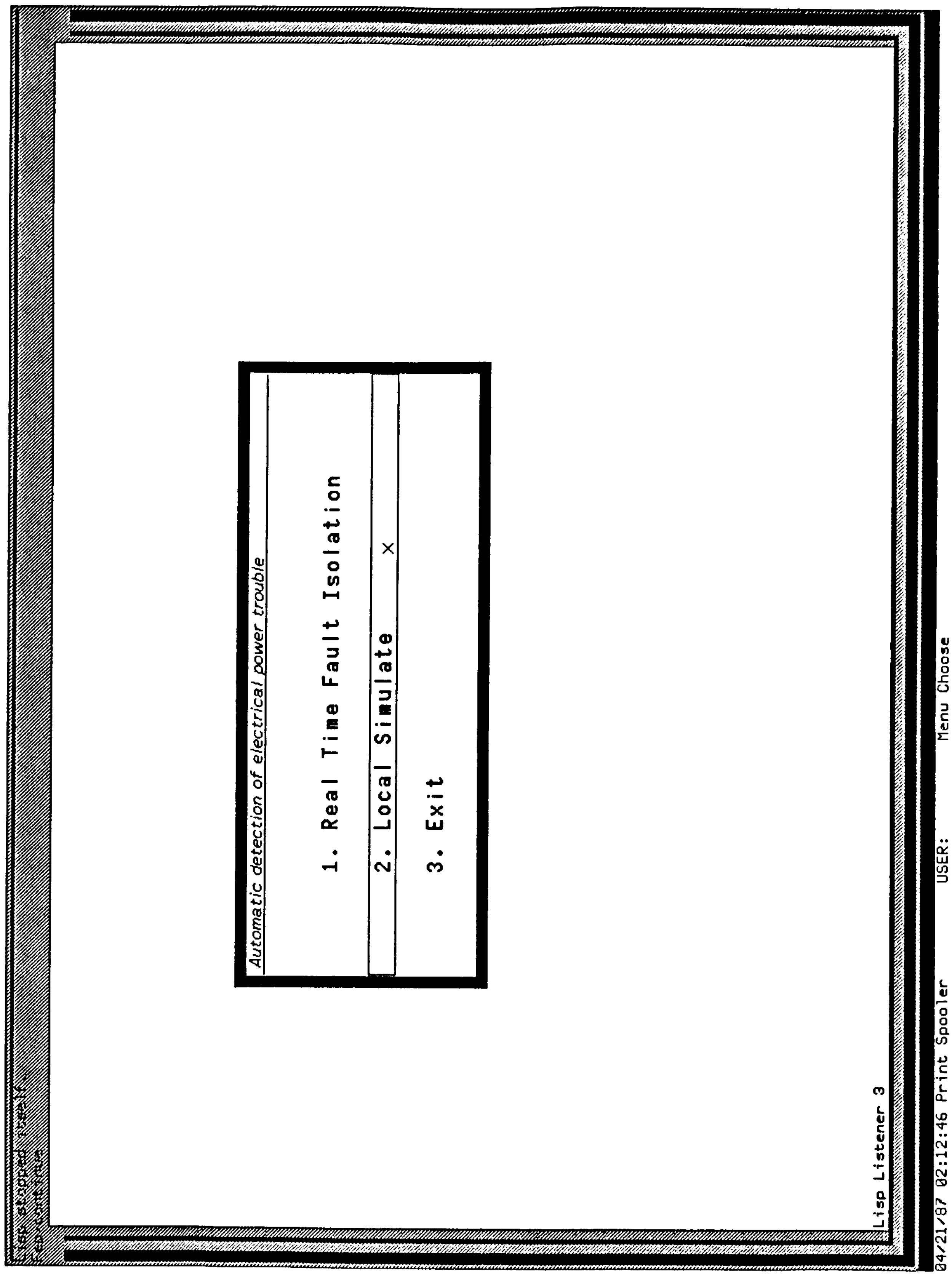
R = 1 S = 1 S.V. = 19.687449 F.V = 5.312486 S.C. = 0.48828 F.C. = 0.48828

R = 2 S = 2 S.V. = 19.687449 F.V = 5.312486 S.C. = 0.48828 F.C. = 0.48828

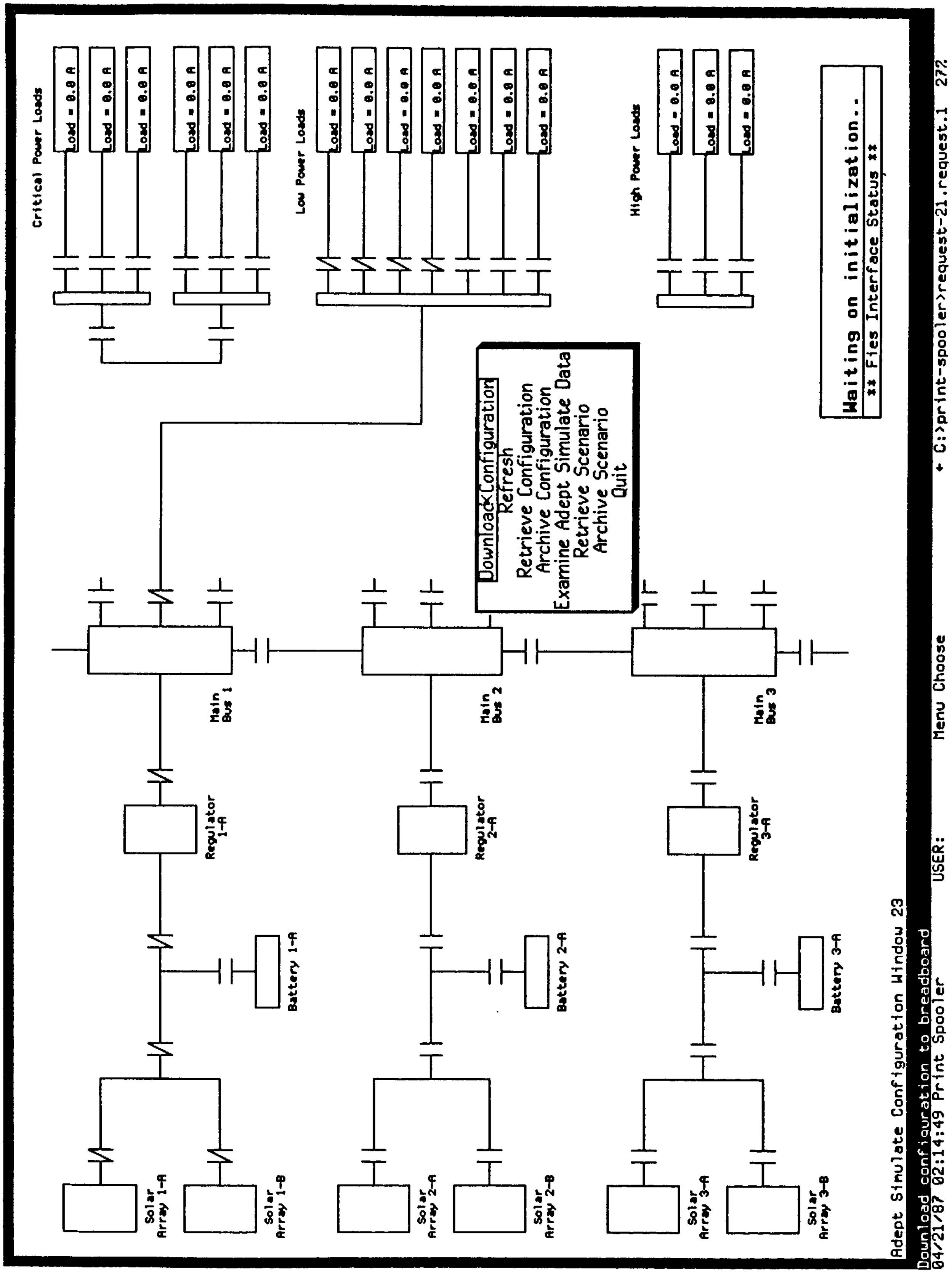
R = 4 S = 4 S.V. = 18.749952 F.V = 4.3749886 S.C. = 1.074216 F.C. = 1.074216

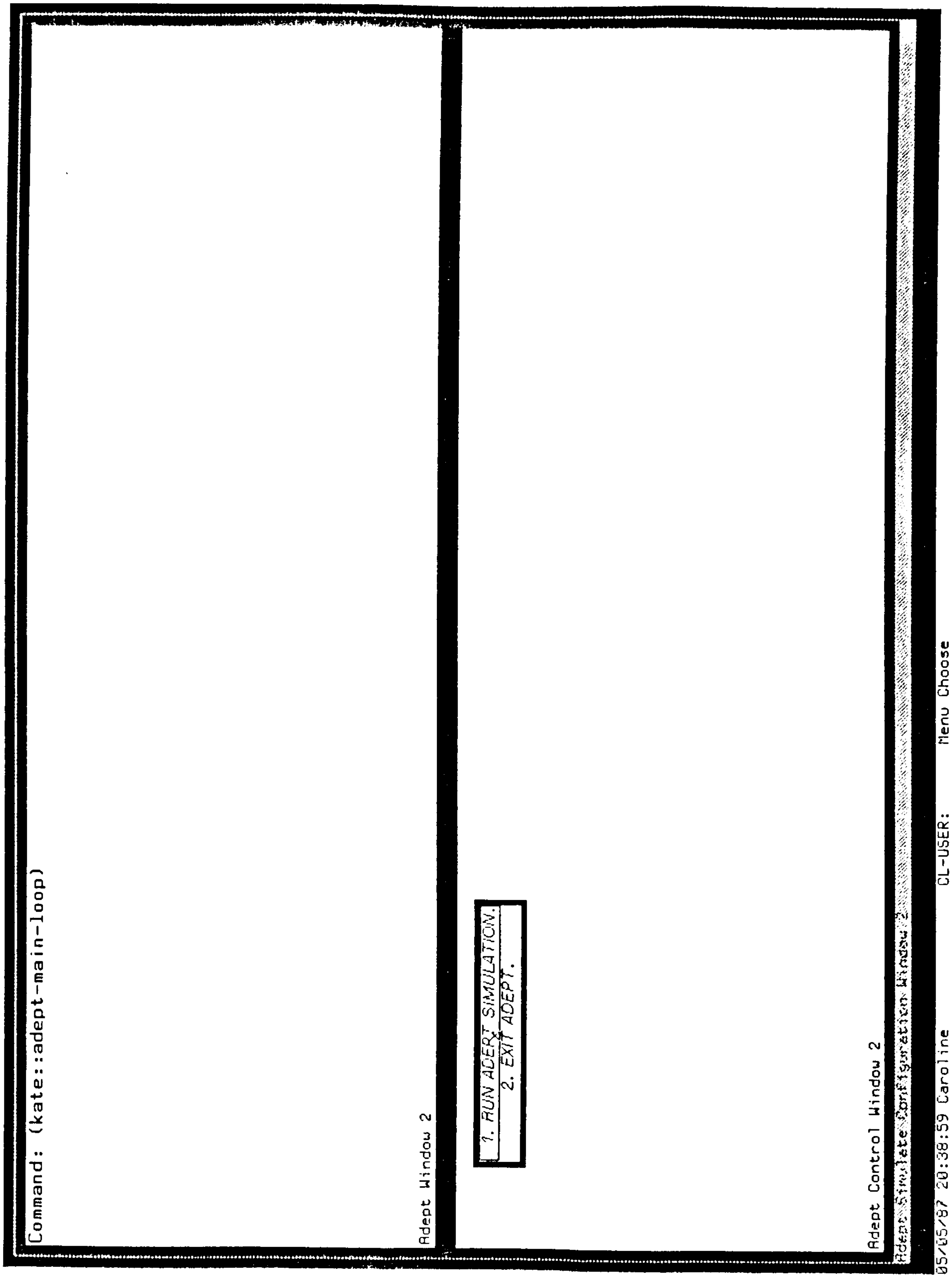
R = 5 S = 5 S.V. = 17.812454 F.V = 3.4374912 S.C. = 0.781248 F.C. = 1.074216

R = 6 S = 6 S.V. = 15.312461 F.V = 0.6249984 S.C. = 0.878904 F.C. = 0.0


Lisp stopped itself.
Automatic detection of electrical power trouble
1. Real Time Fault Isolation
2. Local Simulate
3. Exit
Lisp Listener 3
04/21/87 02:12:46 Print Spooler
USER:
Menu Choose

ORIGINAL PAGE IS
OF POOR QUALITY



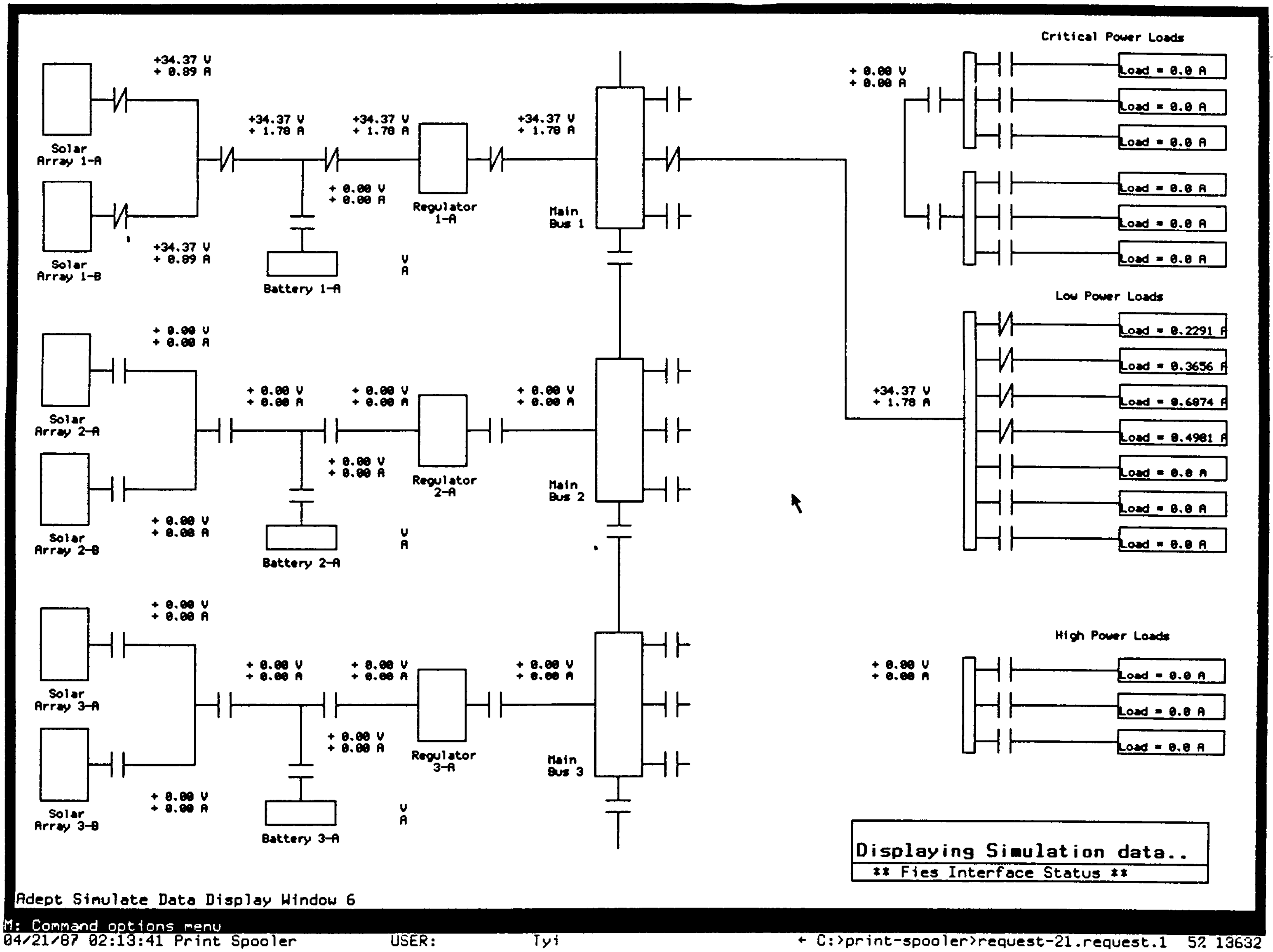

Command: (kate::adept-main-loop)
Adept Window 2
1. RUN ADEPT SIMULATION.
2. EXIT ADEPT.
Adept Control Window 2
Adept Simulate Configuration Window 2
05/05/87 20:38:59 Caroline
CL-USER:
Menu Choose

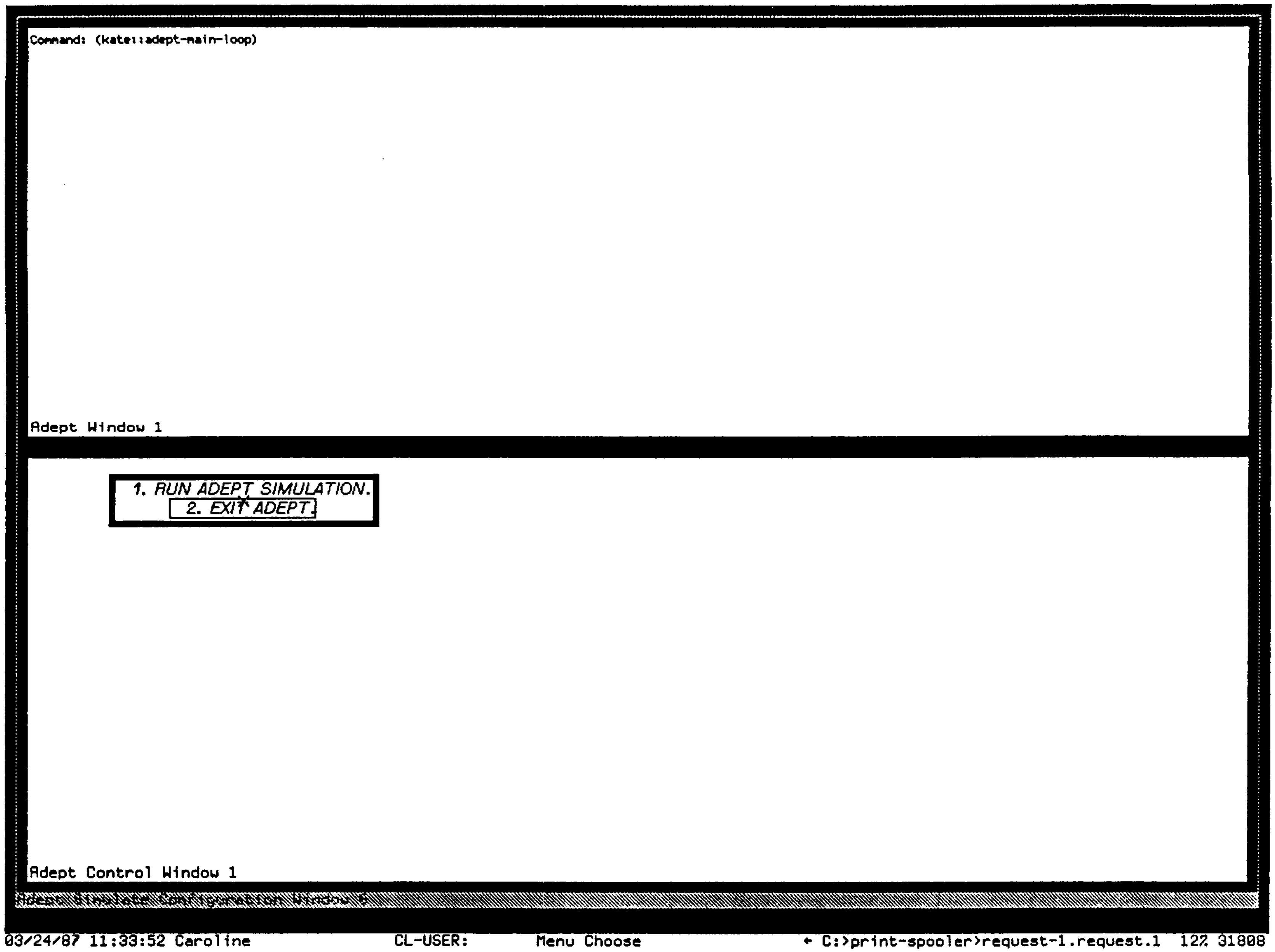
```
Command: (kate::adept-main-loop)C1 ON
VM1 34.37
C2 ON
VM2 34.37
C4 ON
VM4 34.37
C5 ON
VM5 34.37
C6 ON
VM6 34.37
C27 ON
VM20 34.37
C42 ON
AM20 0.22913332
AM1 0.11456666
AM2 0.11456666
AM4 0.22913332
AM5 0.22913332
AM6 0.22913332
C43 ON

Adept Window 3

(kate::goal 'kate::c1 'on t)Turning C1, Solar Array S1-A Select ON.
(kate::goal 'kate::c2 'on t)Turning C2, Solar Array S1-B Select ON.
(kate::goal 'kate::c4 'on t)Turning C4, Solar Array Connect ON.
(kate::goal 'kate::c5 'on t)Turning C5, Power to Regulator R1-A Connect ON.
(kate::goal 'kate::c6 'on t)Turning C6, Regulator R1-A to Main Bus 1 Connect ON.
(kate::goal 'kate::c27 'on t)Turning C27, Main Bus 1 to Low Power Loads Connect ON.
(kate::goal 'kate::c42 'on t)Turning C42, Low Power Load 1 Connect ON.
(kate::goal 'kate::c43 'on t)Turning C43, Low Power Load 2 Connect ON.
(kate::goal 'kate::c44 'on t)Turning C44, Low Power Load 3 Connect ON.

Adept Control Window 3
Adept Simulate Configuration Window 3
04/22/87 11:35:31 Caroline          KATE:        Run          → C:>print-spooler>request-2.proto-request  0
```


Critical Power Loads
Low Power Loads
High Power Loads
Load = 0.0 A
Load = 0.2291 A
Load = 0.3656 A
Load = 0.6874 A
Load = 0.4981 A
+34.37 V
+ 0.89 A
+34.37 V
+ 1.78 A
+ 0.00 V
+ 0.00 A
Solar Array 1-A
Solar Array 1-B
Solar Array 2-A
Solar Array 2-B
Solar Array 3-A
Solar Array 3-B
Battery 1-A
Battery 2-A
Battery 3-A
Regulator 1-A
Regulator 2-A
Regulator 3-A
Main Bus 1
Main Bus 2
Main Bus 3
Displaying Simulation data..
** Fies Interface Status **
Adept Simulate Data Display Window 6
M: Command options menu
04/21/87 02:13:41 Print Spooler USER: Tyi + C:>print-spooler>request-21.request.1 5% 13632


Command: (kate::adept-main-loop)
Adept Window 1
1. RUN ADEPT SIMULATION.
2. EXIT ADEPT.
Adept Control Window 1
Adept Simulate Configuration Window 6
03/24/87 11:33:52 Caroline
CL-USER:
Menu Choose
C:>print-spooler>request-1.request.1 12% 31808

VI. ADEPT RULE SYSTEM

AUTOMATIC DETECTION OF THE ELECTRIC POWER TROUBLE SYSTEMS RULES

Steady state loads on the FIES are resistive. The resistance values for each load are known, therefore the rules for finding inserted faults are based on Ohm's Law. Faults that can be inserted on the FIES are open and closed relays and resistive and direct shunts to ground. On each of the three power busses voltages and currents are measured at optimum points via. Analog to Digital converters.

When an initial configuration of loads is selected and downloaded from the Host Computer, steady state condition is achieved and all the sensor points voltage and currents are read continuously and any significant change at any sensor point indicates a fault has been inserted. This Fault condition is then flagged to the Host computer after which isolation program begins.

FAULT ISOLATION

OPEN CIRCUIT

Open circuit conditions are indicated by a sudden drop in the values of current read at the sensors while the voltage values remain the same or perhaps higher. Isolation is done by searching for a sensor point where the voltage also drops. When this is found the location of the inserted fault lies between the sensor points where there was a voltage and was not a voltage.

DIRECT SHUNT

Direct shunt fault causes a sudden increase in the sensor readings of current values and a decrease in voltage values on sensors nearest the power source. When this occurs the type fault is identified and a search begins for a sensor point where the current reading is nearly zero. The fault is located between the sensor where the current is higher than the steady state current and the one where the current reading is approximately zero.

RESISTOR SHUNT

Resistor shunt causes a sudden increase in current readings on the sensors nearest the power source. A decrease in the voltage may also occur where the load plus the resistor shunt causes the current to exceed the capacity of the solar cells being simulated. Isolation of the resistor shunt fault is done by identifying the first sensor reading with a significant decrease in current. The fault is between this sensor and the last one with the high current reading going back toward the power source.

REFINEMENTS

Refinements in the rules are made using Ohm's Law to further identify the type fault being experienced. This is done by considering the ratio of the values of currents and voltages between steady state and fault conditions.

VII. FUTURE PLANS

The following is an outline of future plans for the ADEPT system.

PLANS

SOFTWARE:

Enhance Fault Diagnosis Techniques
Develop Fault Recovery Techniques
Scheduling

HARDWARE:

Add Color Capability
Add Networking Capability
Add additional system such as Sun system that operates outside of LISP environment.

MILESTONE:

1. ADEPT Real Time Control
Modify KATE to be able to handle FIES control in real time.
(Finishing date Jan. 1, 1988)

2. ADEPT Fault Recovery
(Finishing date April 1, 1988)

3. Automatic Schedule Loads
After fault is found, use new schedule to reconfigure.
(Finishing date July 1, 1988)

4. Hardware Upgrades

a. Symbolics 8-bit high resolution Frame Buffer Color Graphics Terminal with their basic graphics software tool on order. The IBM PC/AT Micro Processor for AI/Expert System Projects.
(Summer 1987)

b. Order a SUN type computer for AI application.
(Fall 1987)

c. Include Symbolics, VAX, SUN and IBM PC/AT on the net work communication system for EB41 AI LAB.

APPENDIX

ADEPT KNOWLEDGE BASE

```
;;; -*- Package: KATE-SIMULATE; Base: 10; Syntax: Common-lisp -*-

;; ADEPT SIMULATE KNOWLEDGE BASE

;; CAROLINE K. WANG
;; EB44
;; (205)-544-3887

(deframe S1-A
        (aio solar-array)
        (source t)
        (source-path t)
        (status 34.37)
        (sinks v1)
        (units volts))

(deframe S1-B
        (aio solar-array)
        (source t)
        (source-path t)
        (status 34.37)
        (sinks v2)
        (units volts))

(deframe b1
        (aio battery)
        (source t)
        (source-path t)
        (status 34.37)
        (sinks v3))

(deframe v1
        (aio voltage)
        (source (ad-cstatus s1-A))
        (source-path (cstatus c1))
        (status (cstatus s1-A))
        (sinks vm1 v4))

(deframe v2
        (aio voltage)
        (source (ad-cstatus s1-B))
        (source-path (cstatus c2))
        (status (cstatus s1-B))
        (sinks vm2 v4))

(deframe psv12
        (aio pseudo-voltage)
        (source t)
        (source-path (or (ad-cstatus v1)
                         (ad-cstatus v2)))
        (status (max (cstatus v1)
                     (cstatus v2)))
        (in-path-of v4))

(deframe v4
        (aio voltage)
        (source (ad-cstatus psv12))
        (source-path (cstatus c4))
        (status (cstatus psv12))
        (sinks vm4 v5))
```

```
(deframe v3
         (aio voltage)
         (source (ad-cstatus b1))
         (source-path (cstatus c3))
         (status (cstatus b1))
         (sinks vm3 v5))

(deframe psv34
         (aio pseudo-voltage)
         (source t)
         (source-path (or (ad-cstatus v3)
                          (ad-cstatus v4)))
         (status (max (cstatus v3)
                      (cstatus v4)))
         (in-path-of v5))

(deframe v5
         (aio voltage)
         (source (ad-cstatus psv34))
         (source-path (cstatus c5))
         (status (cstatus psv34))
         (sinks vm5 r1-a))

(deframe r1-a
         (aio voltage-regulator)
         (source (ad-cstatus v5))
         (source-path t)
         (status (cstatus v5))
         (sinks v6))

(deframe v6
         (aio voltage)
         (source (ad-cstatus r1-a))
         (source-path (cstatus c6))
         (status (cstatus r1-a))
         (sinks vm6 (or v19 v20 v21)))

(deframe v19
         (aio voltage)
         (source (cond ((cstatus c22)
                        (ad-cstatus v6))
                       ((cstatus c23)
                        (ad-cstatus v12))
                       ((cstatus c24)
                        (ad-cstatus v18))))
         (source-path (or (cstatus c22)
                          (cstatus c23)
                          (cstatus c24)))
         (status (cond ((cstatus c22)
                        (cstatus v6))
                       ((cstatus c23)
                        (cstatus v12))
                       ((cstatus c24)
                        (cstatus v18))))
         (sinks vm19 psv19-1 psv19-2))

(deframe psv19-1
         (aio voltage)
         (source (ad-cstatus v19))
         (source-path (cstatus c31))
```

```
        (status (cstatus v19))
        (sinks L1 L2 L3))

(deframe psv19-2
        (aio voltage)
        (source (ad-cstatus v19))
        (source-path (cstatus c32))
        (status (cstatus v19))
        (sinks L4 L5 L6))


;;;; Voltage-kb-channel-2

(deframe S2-A
        (aio solar-array)
        (source t)
        (source-path t)
        (status 34.37)
        (sinks v7)
        (units volts))

(deframe S2-B
        (aio solar-array)
        (source t)
        (source-path t)
        (status 34.37)
        (sinks v8)
        (units volts))

(deframe b2
        (aio battery)
        (source t)
        (source-path t)
        (status 34.37)
        (sinks v9))

(deframe v7
        (aio voltage)
        (source (ad-cstatus s2-A))
        (source-path (cstatus c7))
        (status (cstatus s2-A))
        (sinks vm7 v10))

(deframe v8
        (aio voltage)
        (source (ad-cstatus s2-B))
        (source-path (cstatus c8))
        (status (cstatus s2-B))
        (sinks vm8 v10))

(deframe psv78
        (aio pseudo-voltage)
        (source t)
        (source-path (or (ad-cstatus v7)
                         (ad-cstatus v8)))
        (status (max (cstatus v7)
                     (cstatus v8)))
        (in-path-of v10))
```

```
(deframe v10
        (aio voltage)
        (source (ad-cstatus psv78))
        (source-path (cstatus c10))
        (status (cstatus psv78))
        (sinks vm10 v11))

(deframe v9
        (aio voltage)
        (source (ad-cstatus b2))
        (source-path (cstatus c3))
        (status (cstatus b1))
        (sinks vm3 v5))

(deframe psv9-10
        (aio pseudo-voltage)
        (source t)
        (source-path (or (ad-cstatus v9)
                         (ad-cstatus v10)))
        (status (max (cstatus v9)
                     (cstatus v10)))
        (in-path-of v11))

(deframe v11
        (aio voltage)
        (source (ad-cstatus psv9-10))
        (source-path (cstatus c11))
        (status (cstatus psv9-10))
        (sinks vm11 r2-a))

(deframe r2-a
        (aio voltage-regulator)
        (source (ad-cstatus v11))
        (source-path t)
        (status (cstatus v11))
        (sinks v12))

(deframe v12
        (aio voltage)
        (source (ad-cstatus r2-a))
        (source-path (cstatus c6))
        (status (cstatus r2-a))
        (sinks vm12 (or v19 v20 v21)))

(deframe v20
        (aio voltage)
        (source (cond ((cstatus c27)
                       (ad-cstatus v6))
                      ((cstatus c25)
                       (ad-cstatus v12))
                      ((cstatus c26)
                       (ad-cstatus v18))))
        (source-path (or (cstatus c27)
                         (cstatus c25)
                         (cstatus c26)))
        (status (cond ((cstatus c27)
                       (cstatus v6))
                      ((cstatus c25)
                       (cstatus v12))
                      ((cstatus c26)
```

```
                (cstatus v18))))
        (sinks vm20 L7 L8 L9 L10 L11 L12 L13))

;;;;; Voltage-kb-channel-3

(deframe S3-A
        (aio solar-array)
        (source t)
        (source-path t)
        (status 34.37)
        (sinks v13)
        (units volts))

(deframe S3-B
        (aio solar-array)
        (source t)
        (source-path t)
        (status 34.37)
        (sinks v14)
        (units volts))

(deframe b3
        (aio battery)
        (source t)
        (source-path t)
        (status 34.37)
        (sinks v15))

(deframe v13
        (aio voltage)
        (source (ad-cstatus s3-A))
        (source-path (cstatus c13))
        (status (cstatus s3-A))
        (sinks vm13 v16))

(deframe v14
        (aio voltage)
        (source (ad-cstatus s3-B))
        (source-path (cstatus c14))
        (status (cstatus s3-B))
        (sinks vm14 v16))

(deframe psv13-14
        (aio pseudo-voltage)
        (source t)
        (source-path (or (ad-cstatus v13)
                    (ad-cstatus v14)))
        (status (max (cstatus v13)
                (cstatus v14)))
        (in-path-of v16))

(deframe v16
        (aio voltage)
        (source (ad-cstatus psv13-14))
        (source-path (cstatus c16))
        (status (cstatus psv13-14))
        (sinks vm16 v17))

(deframe v15
```

```
            (aio voltage)
            (source (ad-cstatus b3))
            (source-path (cstatus c15))
            (status (cstatus b3))
            (sinks vm15 v17))

(deframe psv15-16
            (aio pseudo-voltage)
            (source t)
            (source-path (or (ad-cstatus v15)
                             (ad-cstatus v16)))
            (status (max (cstatus v15)
                         (cstatus v16)))
            (in-path-of v17))

(deframe v17
            (aio voltage)
            (source (ad-cstatus psv15-16))
            (source-path (cstatus c17))
            (status (cstatus psv15-16))
            (sinks vm17 r3-a))

(deframe r3-a
            (aio voltage-regulator)
            (source (ad-cstatus v17))
            (source-path t)
            (status (cstatus v17))
            (sinks v18))

(deframe v18
            (aio voltage)
            (source (ad-cstatus r3-a))
            (source-path (cstatus c18))
            (status (cstatus r3-a))
            (sinks vm18 (or v19 v20 v21)))

(deframe v21
            (aio voltage)
            (source (cond ((cstatus c30)
                           (ad-cstatus v6))
                          ((cstatus c28)
                           (ad-cstatus v12))
                          ((cstatus c29)
                           (ad-cstatus v18))))
            (source-path (or (cstatus c30)
                             (cstatus c28)
                             (cstatus c29)))
            (status (cond ((cstatus c30)
                           (cstatus v6))
                          ((cstatus c28)
                           (cstatus v12))
                          ((cstatus c29)
                           (cstatus v18))))
            (sinks vm21 L14 L15 L16))
```

```
;;; -*- Package: KATE-SIMULATE; Base: 10; Syntax: Common-lisp -*-
;
; This file contains the Kate version of Channel one (1) of the fies relay knowledge base
;
(deframe C1
        (nomenclature "Solar Array S1-A Select")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v1 v4)
        (cvalue off))

(deframe C2
        (nomenclature "Solar Array S1-B Select")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v2 v4)
        (cvalue off))

(deframe C3
        (nomenclature "Battery B1-A Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v3)
        (cvalue off))

(deframe C4
        (nomenclature "Solar Array Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v4)
        (cvalue off))

(deframe C5
        (nomenclature "Power to Regulator R1-A Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v5)
        (cvalue off))

(deframe C6
        (nomenclature "Regulator R1-A to Main Bus 1 Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v6)
        (cvalue off))

(deframe C22
        (nomenclature "Main Bus 1 to Critical Loads Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v19)
        (cvalue off))
```

```
(deframe C27
        (nomenclature "Main Bus 1 to Low Power Loads Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v20)
        (cvalue off))

(deframe C30
        (nomenclature "Main Bus 1 to High Power Loads Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v20)
        (cvalue off))

(deframe C31
        (nomenclature "Critical Power Loads Primary System Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of psv19-1)
        (cvalue off))

(deframe C32
        (nomenclature "Critical Power Loads Secondary System Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of psv19-2)
        (cvalue off))

(deframe C33
        (nomenclature "Critical Power Load 1 Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of L1)
        (cvalue off))

(deframe C34
        (nomenclature "Critical Power Load 2 Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of L2)
        (cvalue off))

(deframe C35
        (nomenclature "Critical Power Load 3 Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of L3)
        (cvalue off))

(deframe C36
        (nomenclature "Critical Power Load 4 Connect")
        (aio discrete-command)
        (source t)
```

```
        (source-path t)
        (in-path-of L4)
        (cvalue off))

(deframe C37
        (nomenclature "Critical Power Load 5 Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of L5)
        (cvalue off))

(deframe C38
        (nomenclature "Critical Power Load 6 Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of L6)
        (cvalue off))
;;;;; Channel-2


(deframe C7
        (nomenclature "Solar Array S2-A Select")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v7 v10)
        (cvalue off))

(deframe C8
        (nomenclature "Solar Array S2-B Select")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v8 v10)
        (cvalue off))

(deframe C9
        (nomenclature "Battery B2-A Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v9)
        (cvalue off))

(deframe C10
        (nomenclature "Solar Array Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v10)
        (cvalue off))

(deframe C11
        (nomenclature "Power to Regulator R2-A Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v11)
```

```
        (cvalue off))

(deframe C12
        (nomenclature "Regulator R2-A to Main Bus 2 Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v12)
        (cvalue off))

(deframe C23
        (nomenclature "Main Bus 2 to Critical Loads Connect")
        (aio discrete-command)
        (source t)
        (source-path t)   ;*********************************************
        (in-path-of v19)  ;on Fies board this is switch 23 to S19.
        (cvalue off))     ;*********************************************

(deframe C25
        (nomenclature "Main Bus 2 to Low Power Loads Connect")
        (aio discrete-command)
        (source t)
        (source-path t)   ;*********************************************
        (in-path-of v20)  ;on Fies board this is switch 25 to S20.
        (cvalue off))     ;*********************************************

(deframe C28
        (nomenclature "Main Bus 2 to High Power Loads Connect")
        (aio discrete-command)
        (source t)
        (source-path t)   ;*********************************************
        (in-path-of v21)  ;on Fies board this is switch 28 to S21.
        (cvalue off))     ;*********************************************

;(deframe C20
;       (nomenclature "Low Power Load Systems Connect")
;       (aio discrete-command)
;       (source t)
;       (source-path t)   ;**************************************************
;       (in-path-of psv20) ; This may present a problem. What exactly is C20.
;       (cvalue off))     ;**************************************************

(deframe C42
        (nomenclature "Low Power Load 1 Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of L7)
        (cvalue off))

(deframe C43
        (nomenclature "Low Power Load 2 Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of L8)
        (cvalue off))

(deframe C44
        (nomenclature "Low Power Load 3 Connect")
```

```
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of L9)
        (cvalue off))

(deframe C45
        (nomenclature "Low Power Load 4 Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of L10)
        (cvalue off))

(deframe C46
        (nomenclature "Low Power Load 5 Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of L11)
        (cvalue off))

(deframe C47
        (nomenclature "Low Power Load 6 Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of L12)
        (cvalue off))

(deframe C48
        (nomenclature "Low Power Load 7 Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of L13)
        (cvalue off))


; This file contains the Kate version of Channel three (3) of the Fies relay knowledge base.
;
(deframe C13
        (nomenclature "Solar Array S3-A Select")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v13 v16)
        (cvalue off))

(deframe C14
        (nomenclature "Solar Array S3-B Select")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v14 v16)
        (cvalue off))

(deframe C15
        (nomenclature "Battery B3-A Connect")
        (aio discrete-command)
```

```
        (source t)
        (source-path t)
        (in-path-of v15)
        (cvalue off))

(deframe C16
        (nomenclature "Solar Array Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v16)
        (cvalue off))

(deframe C17
        (nomenclature "Power to Regulator R3-A Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v17)
        (cvalue off))

(deframe C18
        (nomenclature "Regulator R-A to Main Bus 3 Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v18)
        (cvalue off))

(deframe C24
        (nomenclature "Main Bus 3 to Critical Loads Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v19)
        (cvalue off))

(deframe C26
        (nomenclature "Main Bus 3 to Low Power Loads Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v20)
        (cvalue off))

(deframe C29
        (nomenclature "Main Bus 3 to High Power Loads Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of v21)
        (cvalue off))

;(deframe C21
;       (nomenclature "High Power Load Systems Connect")
;       (aio discrete-command)
;       (source t)
;       (source-path t)
;       (in-path-of psv21)
;       (cvalue off))
```

```
(deframe C39
        (nomenclature "High Power Load 1 Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of L14)
        (cvalue off))

(deframe C40
        (nomenclature "High Power Load 2 Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of L15)
        (cvalue off))

(deframe C41
        (nomenclature "High Power Load 3 Connect")
        (aio discrete-command)
        (source t)
        (source-path t)
        (in-path-of L16)
        (cvalue off))
```

```
;;; -*- Package: KATE-SIMULATE; Base: 10; Syntax: Common-lisp -*-

; Update by SDavis toward Fies Interface. This is Load 2 (low power loads).
; Note that all psn-2 have been changed to psn-1's.
; Note that all resistance's have been changed also.
;
;*************************************************************************************
******
; Critical Power Loads   USING:  psv19-1 & psv19-2 , psa19-1 & psa19-2 nodes
;*************************************************************************************
******

(deframe L1
        (aio load)
        (source (ad-cstatus psv19-1))
        (source-path (cstatus c33))
        (status (/ (cstatus psv19-1) 100.))
        (in-path-of psa19-1)
        (units "amperes"))

(deframe L2
        (aio load)
        (source (ad-cstatus psv19-1))
        (source-path (cstatus c34))
        (status (/ (cstatus psv19-1) 50.))
        (in-path-of psa19-1)
        (units "amperes"))
(deframe L3
        (aio load)
        (source (ad-cstatus psv19-1))
        (source-path (cstatus c35))
        (status (/ (cstatus psv19-1) 27.))
        (in-path-of psa19-1)
        (units "amperes"))

(deframe L4
        (aio load)
        (source (ad-cstatus psv19-2))
        (source-path (cstatus c36))
        (status (/ (cstatus psv19-2) 100.))
        (in-path-of psa19-2)
        (units "amperes"))

(deframe L5
        (aio load)
        (source (ad-cstatus psv19-2))
        (source-path (cstatus c37))
        (status (/ (cstatus psv19-2) 50.))
        (in-path-of psa19-2)
        (units "amperes"))

(deframe L6
        (aio load)
        (source (ad-cstatus psv19-2))
        (source-path (cstatus c38))
        (status (/ (cstatus psv19-2) 27.))
        (in-path-of psa19-2)
        (units "amperes"))

;*************************************************************************************
```

```
******
; Low Power Loads    USING  psv20-1 , psa20-1 nodes
;*******************************************************************************
******

(deframe L7
        (aio load)
        (source (ad-cstatus v20))
        (source-path (cstatus c42))
        (status (/ (cstatus v20) 150.))
        (in-path-of a20)
        (units "amperes"))

(deframe L8
        (aio load)
        (source (ad-cstatus v20))
        (source-path (cstatus c43))
        (status (/ (cstatus v20) 94.))
        (in-path-of a20)
        (units "amperes"))

(deframe L9
        (aio load)
        (source (ad-cstatus v20))
        (source-path (cstatus c44))
        (status (/ (cstatus v20) 50.))
        (in-path-of a20)
        (units "amperes"))

(deframe L10
        (aio load)
        (source (ad-cstatus v20))
        (source-path (cstatus c45))
        (status (/ (cstatus v20) 69.))
        (in-path-of a20)
        (units "amperes"))

(deframe L11
        (aio load)
        (source (ad-cstatus v20))
        (source-path (cstatus c46))
        (status (/ (cstatus v20) 74.))
        (in-path-of a20)
        (units "amperes"))

(deframe L12
        (aio load)
        (source (ad-cstatus v20))
        (source-path (cstatus c47))
        (status (/ (cstatus v20) 50.))
        (in-path-of a20)
        (units "amperes"))
(deframe L13
        (aio load)
        (source (ad-cstatus v20))
        (source-path (cstatus c48))
```

```
        (status (/ (cstatus v20) 81.))
        (in-path-of a20)
        (units "amperes"))


;**************************************************************************************
******
; High Power Loads    USING  v21 , a21 nodes
;**************************************************************************************
******


(deframe L14
        (aio load)
        (source (ad-cstatus v21))
        (source-path (cstatus c39))
        (status (/ (cstatus v21) 27.))
        (in-path-of a21)
        (units "amperes"))

(deframe L15
        (aio load)
        (source (ad-cstatus v21))
        (source-path (cstatus c40))
        (status (/ (cstatus v21) 14.))
        (in-path-of a21)
        (units "amperes"))

(deframe L16
        (aio load)
        (source (ad-cstatus v21))
        (source-path (cstatus c41))
        (status (/ (cstatus v21) 14.))
        (in-path-of a21)
        (units "amperes"))
```

```lisp
;;; -*- Package: KATE-SIMULATE; Base: 10; Syntax: Common-lisp -*-

(deframe voltage
	(ako analog-object))

(deframe voltage-measurement
	(ako analog-measurement))

(deframe pseudo-voltage
	(ako analog-object))

(deframe passive-item)

(deframe analog-passive-item
	(ako analog-object passive-item))

(deframe current
	(ako analog-passive-item))

(deframe current-measurement
	(ako analog-passive-item))

(deframe pseudo-current
	(ako analog-passive-item))

(deframe voltage-regulator
	(ako analog-object))

(deframe solar-array
	(ako analog-object))

(deframe battery
	(ako analog-object))

(deframe load
	(ako analog-object analog-passive-item))
```

```
;;; -*- Package: KATE-SIMULATE; Base: 10; Syntax: Common-lisp -*-

(deframe am19
          (aio current-measurement)
          (source (ad-cstatus a19))
          (source-path t)
          (status (cstatus a19))
          (cvalue 0.0)
          (units "amperes"))

(deframe am6
          (aio current-measurement)
          (source (ad-cstatus a6))
          (source-path t)
          (status (cstatus a6))
          (cvalue 0.0)
          (units "amperes"))

(deframe am5
          (aio current-measurement)
          (source (ad-cstatus a5))
          (source-path t)
          (status (cstatus a5))
          (cvalue 0.0)
          (units "amperes"))

(deframe am4
          (aio current-measurement)
          (source (ad-cstatus a4))
          (source-path t)
          (status (cstatus a4))
          (cvalue 0.0)
          (units "amperes"))

(deframe am3
          (aio current-measurement)
          (source (ad-cstatus a3))
          (source-path t)
          (status (cstatus a3))
          (cvalue 0.0)
          (units "amperes"))

(deframe am2
          (aio current-measurement)
          (source (ad-cstatus a2))
          (source-path t)
          (status (cstatus a2))
          (cvalue 0.0)
          (units "amperes"))

(deframe am1
          (aio current-measurement)
          (source (ad-cstatus a1))
          (source-path t)
          (status (cstatus a1))
          (cvalue 0.0)
          (units "amperes"))
```

```
;;;;;; Current-measurement-2

(deframe am20
        (aio current-measurement)
        (source (ad-cstatus a20))
        (source-path t)
        (status (cstatus a20))
        (cvalue 0.0)
        (units "amperes"))

(deframe am12
        (aio current-measurement)
        (source (ad-cstatus a12))
        (source-path t)
        (status (cstatus a12))
        (cvalue 0.0)
        (units "amperes"))

(deframe am11
        (aio current-measurement)
        (source (ad-cstatus a11))
        (source-path t)
        (status (cstatus a11))
        (cvalue 0.0)
        (units "amperes"))

(deframe am10
        (aio current-measurement)
        (source (ad-cstatus a10))
        (source-path t)
        (status (cstatus a10))
        (cvalue 0.0)
        (units "amperes"))

(deframe am9
        (aio current-measurement)
        (source (ad-cstatus a9))
        (source-path t)
        (status (cstatus a9))
        (cvalue 0.0)
        (units "amperes"))

(deframe am8
        (aio current-measurement)
        (source (ad-cstatus a8))
        (source-path t)
        (status (cstatus a8))
        (cvalue 0.0)
        (units "amperes"))

(deframe am8
        (aio current-measurement)
        (source (ad-cstatus a7))
        (source-path t)
        (status (cstatus a7))
        (cvalue 0.0)
        (units "amperes"))
```

```
;;;;;;Current-measurement-3

(deframe am21
        (aio current-measurement)
        (source (ad-cstatus a21))
        (source-path t)
        (status (cstatus a21))
        (cvalue 0.0)
        (units "amperes"))

(deframe am18
        (aio current-measurement)
        (source (ad-cstatus a18))
        (source-path t)
        (status (cstatus a18))
        (cvalue 0.0)
        (units "amperes"))

(deframe am17
        (aio current-measurement)
        (source (ad-cstatus a17))
        (source-path t)
        (status (cstatus a17))
        (cvalue 0.0)
        (units "amperes"))

(deframe am16
        (aio current-measurement)
        (source (ad-cstatus a16))
        (source-path t)
        (status (cstatus a16))
        (cvalue 0.0)
        (units "amperes"))

(deframe am15
        (aio current-measurement)
        (source (ad-cstatus a15))
        (source-path t)
        (status (cstatus a15))
        (cvalue 0.0)
        (units "amperes"))

(deframe am14
        (aio current-measurement)
        (source (ad-cstatus a14))
        (source-path t)
        (status (cstatus a14))
        (cvalue 0.0)
        (units "amperes"))

(deframe am13
        (aio current-measurement)
        (source (ad-cstatus a13))
        (source-path t)
        (status (cstatus a13))
        (cvalue 0.0)


        (units "amperes"))
```

```
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: KATE-SIMULATE; Base: 10 -*-


(deframe psA19-1
         (aio current)
         (source t)
         (source-path (or (ad-cstatus L1)
                          (ad-cstatus L2)
                          (ad-cstatus L3)))
         (status (+ (cstatus L1)
                    (cstatus L2)
                    (cstatus L3)))
         (in-path-of a19))

(deframe psA19-2
         (aio current)
         (source t)
         (source-path (or (ad-cstatus L4)
                          (ad-cstatus L5)
                          (ad-cstatus L6)))
         (status (+ (cstatus L4)
                    (cstatus L5)
                    (cstatus L6)))
         (in-path-of a19))


(deframe A19
         (aio current)              ;this is an unusual way to represent current
         (source t)           ;but more than one source is not allowed at this
         (source-path (or (ad-cstatus psa19-1)  ;time and this representation does work.
                          (ad-cstatus psa19-2)))
         (status (+ (cstatus psa19-1)
                    (cstatus psa19-2)))
         (sinks am19a a6 a12 a18))


(deframe A6
         (aio current)
         (source (cond ((cstatus c22)
                        (ad-cstatus a19))
                       ((cstatus c27)
                        (ad-cstatus a20))
                       ((cstatus c30)
                        (ad-cstatus a21))))
         (source-path t)
         (status (cond ((cstatus c22)
                        (cstatus a19))
                       ((cstatus c27)
                        (cstatus a20))
                       ((cstatus c30)
                        (cstatus a21))))
         (sinks a5 am6))

(deframe A5
         (aio current)
         (source (ad-cstatus a6))
         (source-path (cstatus c6))
         (status (cstatus a6))
```

```
        (sinks a3 a4 am5))

(deframe A3
        (aio current)
        (source (ad-cstatus a5))
        (source-path t)
              ; The status equation assumes that the resistance
              ; for the solar arrays and battery are equal and
              ; that the battery is sharing the load when it is
              ; connected with the arrays. This probably is not the
              ; case. Therefore this equation will have to change
              ; to better model the real hardware.
        (status (cond ((not (cstatus c3)) 0.0)
                      ((not (cstatus c4)) (cstatus a5))
                      ((and (cstatus c1) (cstatus c2))
                       (/ (cstatus a5) 3.0))
                      ((or (cstatus c1) (cstatus c2))
                       (/ (cstatus a5) 2.0))
                      (t 0.0)))
        (sinks a4 am3))

(deframe A4
        (aio current)
        (source (ad-cstatus a5))
        (source-path t)
        (status (- (cstatus a5)
                   (cstatus a3)))
        (sinks a1 a2 am4))

(deframe A2
        (aio current)
        (source (ad-cstatus a4))
        (source-path t)
        (status (cond ((and (cstatus c1) (cstatus c2))
                       (/ (cstatus a4) 2.))
                      ((and (cstatus c2) (not (cstatus c1)))
                       (cstatus a4))
                      (t 0.0)))
        (sinks am2))

(deframe A1
        (aio current)
        (source (ad-cstatus a4))
        (source-path t)
        (status (- (cstatus a4)
                   (cstatus a2)))
        (sinks am1))


;;;;; Current-kb-2


(deframe A20
        (aio current)          ;this is an unusual way to represent current
        (source t)          ;but more than one source is not allowed at this
        (source-path (or (ad-cstatus L7)
                         (ad-cstatus L8)
                         (ad-cstatus L9)
                         (ad-cstatus L10)
                         (ad-cstatus L11)
```

```
            (ad-cstatus L12)
            (ad-cstatus L13)))
    (status (+ (cstatus L7)
          (cstatus L8)
          (cstatus L9)
          (cstatus L10)
          (cstatus L11)
          (cstatus L12)
          (cstatus L13)))
    (sinks am20 a6 a12 a18))


(deframe A12
    (aio current)
    (source (cond ((cstatus c23)
             (ad-cstatus a19))
            ((cstatus c25)
             (ad-cstatus a20))
            ((cstatus c28)
             (ad-cstatus a21))))
    (source-path t)
    (status (cond ((cstatus c23)
             (cstatus a19))
            ((cstatus c25)
             (cstatus a20))
            ((cstatus c28)
             (cstatus a21))))
    (sinks a11 am12))

(deframe A11
    (aio current)
    (source (ad-cstatus a12))
    (source-path (cstatus c12))
    (status (cstatus a12))
    (sinks a9 a10 am11))

(deframe A9
    (aio current)
    (source (ad-cstatus a11))
    (source-path t)
        ; The status equation assumes that the resistance
        ; for the solar arrays and battery are equal and
        ; that the battery is sharing the load when it is
        ; connected with the arrays. This probably is not the
        ; case. Therefore this equation will have to change
        ; to better model the real hardware.
    (status (cond ((not (cstatus c9)) 0.0)
            ((not (cstatus c10)) (cstatus a11))
            ((and (cstatus c7) (cstatus c8))
             (/ (cstatus a11) 3.0))
            ((or (cstatus c7) (cstatus c8))
             (/ (cstatus a11) 2.0))
            (t 0.0)))
    (sinks a10 am9))

(deframe A10
    (aio current)
    (source (ad-cstatus a11))
    (source-path t)
    (status (- (cstatus a11)
```

```
        (cstatus a9)))
    (sinks a7 a8 am10))

(deframe A8
    (aio current)
    (source (ad-cstatus a10))
    (source-path t)
    (status (cond ((and (cstatus c1) (cstatus c2))
                   (/ (cstatus a10) 2.))
                  ((and (cstatus c8) (not (cstatus c7)))
                   (cstatus a10))
                  (t 0.0)))
    (sinks am8))

(deframe A7
    (aio current)
    (source (ad-cstatus a10))
    (source-path t)
    (status (- (cstatus a10
            (cstatus a8))))
    (sinks am7))


;;;;; Current-kb-3

(deframe A21
    (aio current)           ;this is an unusual way to represent current
    (source t)          ;but more than one source is not allowed at this
    (source-path (or (ad-cstatus L14)
                     (ad-cstatus L15)
                     (ad-cstatus L16)))
    (status (+ (cstatus L14)
            (cstatus L15)
            (cstatus L16)))
    (sinks am21 a6 a12 a18))


(deframe A18
    (aio current)
    (source (cond ((cstatus c24)
                   (ad-cstatus a19))
                  ((cstatus c26)
                   (ad-cstatus a20))
                  ((cstatus c29)
                   (ad-cstatus a21))))

    (source-path t)
    (status (cond ((cstatus c24)
                   (cstatus a19))
                  ((cstatus 26)
                   (cstatus a20))
                  ((cstatus a29)
                   (cstatus a21))))
    (sinks a17 am18))

(deframe A17
    (aio current)
    (source (ad-cstatus a18))
    (source-path (cstatus c18))
    (status (cstatus a18))
```

```
            (sinks a15 a16 am17))

    (deframe A15
            (aio current)
            (source (ad-cstatus a17))
            (source-path t)
                ; The status equation assumes that the resistance
                ; for the solar arrays and battery are equal and
                ; that the battery is sharing the load when it is
                ; connected with the arrays. This probably is not the
                ; case. Therefore this equation will have to change
                ; to better model the real hardware.
            (status (cond ((not (cstatus c15)) 0.0)
                        ((not (cstatus c16)) (cstatus a17))
                        ((and (cstatus c13) (cstatus c14))
                         (/ (cstatus a17) 3.0))
                        ((or (cstatus c13) (cstatus c14))
                         (/ (cstatus a17) 2.0))
                        (t 0.0)))
            (sinks a16 am15))

    (deframe A16
            (aio current)
            (source (ad-cstatus a17))
            (source-path t)
            (status (- (cstatus a17)
                    (cstatus a15)))
            (sinks a13 a14 am16))

    (deframe A14
            (aio current)
            (source (ad-cstatus a16))
            (source-path t)
            (status (cond ((and (cstatus c13) (cstatus c14))
                         (/ (cstatus a16) 2.))
                        ((and (cstatus c14) (not (cstatus c13)))
                         (cstatus a16))
                        (t 0.0)))
            (sinks am14))

    (deframe A13
            (aio current)
            (source (ad-cstatus a16))
            (source-path t)
            (status (- (cstatus a16)
                    (cstatus a14)))
            (sinks am13))
```

```
(deframe S3-A
         (aio solar-array)
         (source t)
         (source-path t)
         (status 34.37)
         (sinks v13)
         (units volts))

(deframe S3-B
         (aio solar-array)
         (source t)
         (source-path t)
         (status 34.37)
         (sinks v14)
         (units volts))

(deframe b3
         (aio battery)
         (source t)
         (source-path t)
         (status 34.37)
         (sinks v15))

(deframe v13
         (aio voltage)
         (source (ad-cstatus s3-A))
         (source-path (cstatus c13))
         (status (cstatus s3-A))
         (sinks vm13 v16))

(deframe v14
         (aio voltage)
         (source (ad-cstatus s3-B))
         (source-path (cstatus c14))
         (status (cstatus s3-B))
         (sinks vm14 v16))

(deframe psv13-14
         (aio pseudo-voltage)
         (source t)
         (source-path (or (ad-cstatus v13)
                          (ad-cstatus v14)))
         (status (max (cstatus v13)
                      (cstatus v14)))
         (in-path-of v16))

(deframe v16
         (aio voltage)
         (source (ad-cstatus psv13-14))
         (source-path (cstatus c16))
         (status (cstatus psv13-14))
         (sinks vm16 v17))

(deframe v15
         (aio voltage)
         (source (ad-cstatus b3))
         (source-path (cstatus c15))
         (status (cstatus b3))
         (sinks vm15 v17))
```

```
(deframe psv15-16
        (aio pseudo-voltage)
        (source t)
        (source-path (or (ad-cstatus v15)
                         (ad-cstatus v16)))
        (status (max (cstatus v15)
                     (cstatus v16)))
        (in-path-of v17))

(deframe v17
        (aio voltage)
        (source (ad-cstatus psv15-16))
        (source-path (cstatus c17))
        (status (cstatus psv15-16))
        (sinks vm17 r3-a))

(deframe r3-a
        (aio voltage-regulator)
        (source (ad-cstatus v17))
        (source-path t)
        (status (cstatus v17))
        (sinks v18))

(deframe v18
        (aio voltage)
        (source (ad-cstatus r3-a))
        (source-path (cstatus c18))
        (status (cstatus r3-a))
        (sinks vm18 (or v19 v20 v21)))

(deframe v21
        (aio voltage)
        (source (cond ((cstatus c30)
                       (ad-cstatus v6))
                      ((cstatus c28)
                       (ad-cstatus v12))
                      ((cstatus c29)
                       (ad-cstatus v18))))
        (source-path (or (cstatus c30)
                         (cstatus c28)
                         (cstatus c29)))
        (status (cond ((cstatus c30)
                       (cstatus v6))
                      ((cstatus c28)
                       (cstatus v12))
                      ((cstatus c29)
                       (cstatus v18))))
        (sinks vm21 L14 L15 L16))
```

```
;;; -*- Package: KATE-SIMULATE; Base: 10; Syntax: Common-lisp -*-

(deframe VM1
	(aio voltage-measurement)
	(source (ad-cstatus v1))
	(source-path t)
	(status (cstatus v1))
	(cvalue 0.0))

(deframe VM2
	(aio voltage-measurement)
	(source (ad-cstatus v2))
	(source-path t)
	(status (cstatus v2))
	(cvalue 0.0))

(deframe VM3
	(aio voltage-measurement)
	(source (ad-cstatus v3))
	(source-path t)
	(status (cstatus v3))
	(cvalue 0.0))

(deframe VM4
	(aio voltage-measurement)
	(source (ad-cstatus v4))
	(source-path t)
	(status (cstatus v4))
	(cvalue 0.0))


(deframe VM5
	(aio voltage-measurement)
	(source (ad-cstatus v5))
	(source-path t)
	(status (cstatus v5))
	(cvalue 0.0))


(deframe VM6
	(aio voltage-measurement)
	(source (ad-cstatus v6))
	(source-path t)
	(status (cstatus v6))
	(cvalue 0.0))

(deframe VM19
	(aio voltage-measurement)
	(source (ad-cstatus v19))
	(source-path t)
	(status (cstatus v19))
	(cvalue 0.0))
;;;; Voltage-measurement-kb-channel-2

(deframe VM7
```

```
        (aio voltage-measurement)
        (source (ad-cstatus v7))
        (source-path t)
        (status (cstatus v7))
        (cvalue 0.0))

(deframe VM8
        (aio voltage-measurement)
        (source (ad-cstatus v8))
        (source-path t)
        (status (cstatus v8))
        (cvalue 0.0))

(deframe VM9
        (aio voltage-measurement)
        (source (ad-cstatus v9))
        (source-path t)
        (status (cstatus v9))
        (cvalue 0.0))

(deframe VM10
        (aio voltage-measurement)
        (source (ad-cstatus v10))
        (source-path t)
        (status (cstatus v10))
        (cvalue 0.0))


(deframe VM11
        (aio voltage-measurement)
        (source (ad-cstatus v11))
        (source-path t)
        (status (cstatus v11))
        (cvalue 0.0))


(deframe VM12
        (aio voltage-measurement)
        (source (ad-cstatus v12))
        (source-path t)
        (status (cstatus v12))
        (cvalue 0.0))

(deframe VM20
        (aio voltage-measurement)
        (source (ad-cstatus v20))
        (source-path t)
        (status (cstatus v20))
        (cvalue 0.0))

;;;; Voltage-measurement-kb-channel-3

(deframe VM13
        (aio voltage-measurement)
        (source (ad-cstatus v13))
```

```
        (source-path t)
        (status (cstatus v13))
        (cvalue 0.0))

(deframe VM14
        (aio voltage-measurement)
        (source (ad-cstatus v14))
        (source-path t)
        (status (cstatus v14))
        (cvalue 0.0))

(deframe VM15
        (aio voltage-measurement)
        (source (ad-cstatus v15))
        (source-path t)
        (status (cstatus v15))
        (cvalue 0.0))

(deframe VM16
        (aio voltage-measurement)
        (source (ad-cstatus v16))
        (source-path t)
        (status (cstatus v16))
        (cvalue 0.0))


(deframe VM17
        (aio voltage-measurement)
        (source (ad-cstatus v17))
        (source-path t)
        (status (cstatus v17))
        (cvalue 0.0))


(deframe VM18
        (aio voltage-measurement)
        (source (ad-cstatus v18))
        (source-path t)
        (status (cstatus v18))
        (cvalue 0.0))

(deframe VM21
        (aio voltage-measurement)
        (source (ad-cstatus v21))
        (source-path t)
        (status (cstatus v21))
        (cvalue 0.0))
```

APPROVAL

AUTOMATIC DETECTION OF ELECTRIC POWER TROUBLES (AI Application)

By Caroline Wang, Hugh Zeanah, Audie Anderson, and Clint Patrick

The information in this report has been reviewed for technicalcontent. Review of any information concerning Department of Defense or nuclear energy activities or programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.

W C Bradford

W. C. BRADFORD
Director, Information and Electronic Systems Laboratory

☆U.S. GOVERNMENT PRINTING OFFICE 1987—730-067/40123